www.elektormagazine.de

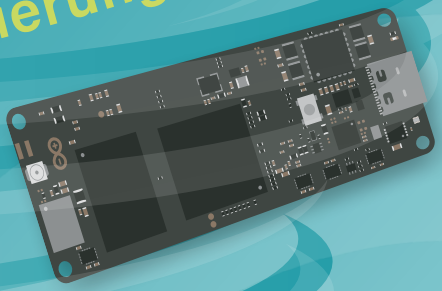
elektorMAG
design > share > earn

Woche 1
DEZ. 2022/JAN. 2023
ELEKTORMAGAZINE.DE
* SEIT 1961 *

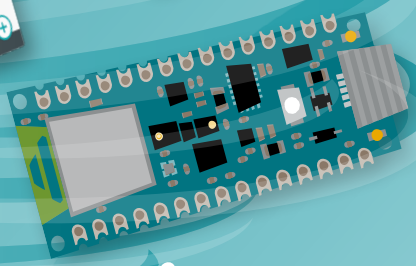
Sonderausgabe
Gastbeiträge von
ARDUINO

Freigeschaltet: die Bonus-Ausgabe!

Hausautomatisierung

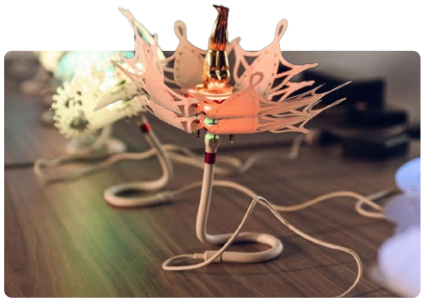


Vernetzung **vereinfacht**



Prototyping für die **Produktion**

Artikel für Profis, Maker und Studenten!



Von Arduino und **Elekor** für **Elektronik-Fans erstellt**

Jetzt Erhältlich

Ein großartiges Elektor Magazine erstellt zusammen mit Arduino!

DIY-Elektronikprojekte, technische Inhalte und vieles mehr von Arduino und Elektor

Viele tolle Projekte und Tutorials

Steigt ein in Inhalte wie MicroPython, TinyML und Heim-automatisierung mit Arduino

Lernt die Arduino Macher kennen: Fabio, Massimo und David

Lernt Alles über das Portenta X8

Ab jetzt in Kiosken, Onlineshops und natürlich beim Elektor Store erhältlich!

In allen Artikeln sind Links zu passenden Lösungen von Arduino gegeben. Somit wird es noch leichter an weitere Infos zu kommen.

Jetzt Kaufen!

www.elektor.de/arduino-magazine

× elektor

53. Jahrgang,
Dezember 2022 / Januar 2023
ISSN 0932-5468
Erscheinungsweise: 8x jährlich

Verlag
Elektor Verlag GmbH
Kackertstraße 10, 52072 Aachen
Tel. 0241 95509190

Technische Fragen: redaktion@elektor.de

Hauptsitz des Verlags
Elektor International Media
Postbus 11, 6114 ZG Susteren (NL)
Niederlande

Anzeigen
Büsra Kas: Tel. 0241 95509178
E-Mail: busra.kas@elektor.com

Distribution
IPS Pressevertrieb GmbH
Postfach 12 11, 53334 Meckenheim

Druck
Senefelder Misset, Doetinchem (NL)

Freigeschaltet: die Bonus-Ausgabe

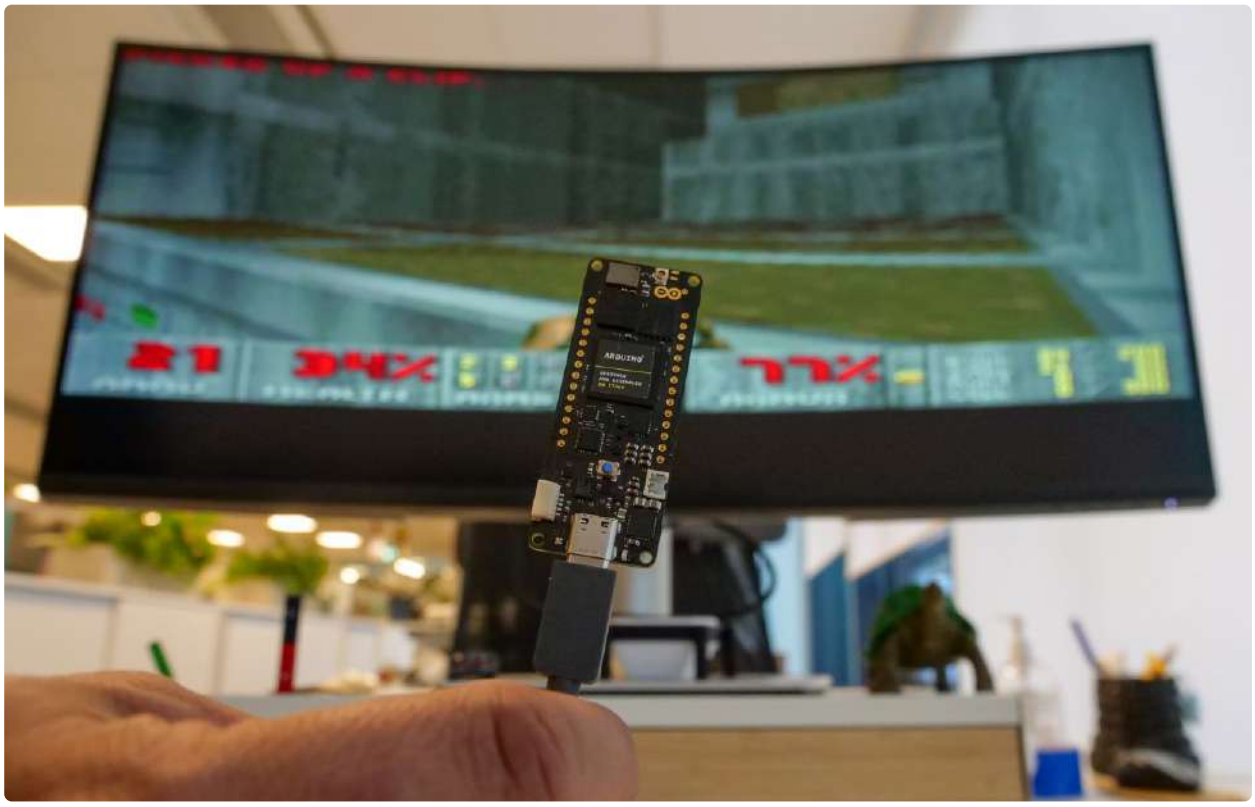
Die kreative Zusammenarbeit zwischen Elektor und Arduino endete nicht mit der Gastausgabe von Elektor, die wir Anfang Dezember 2022 veröffentlichten. Wir haben weitere Projekte, technische Einblicke und informative Artikel für Sie, die Sie in den kommenden Monaten anregen sollen. In den nächsten vier Wochen werden wir die Inhalte dieser Ausgabe für Sie freischalten, bis Sie Anfang Januar 2023 das komplette Bonusheft in Ihren Händen halten. Ist das eine tolle Art, das neue Jahr zu begrüßen?
Ob Sie ein professioneller Ingenieur sind, der an einem neuen Industrieprodukt arbeitet, oder ein Heimwerker auf der Suche nach einem lustigen Wochenendprojekt auf Arduino-Basis, Sie werden diese Extra-Ausgabe von Elektor sicher informativ und inspirierend finden. Wir stellen Ihnen Artikel zu einer Vielzahl von Arduino-bezogenen Themen und Projekten zur Verfügung, darunter Retro-Gaming mit Arduino, ein Arduino-Trainingsboard aus dem Hause Elektor und einen portablen Arduino-basierten Controller für Spotify.
Wenn Sie die Projekte und Artikel in dieser Bonusausgabe lesen, können Sie uns gerne Ihre Gedanken auf elektormagazine.de, arduino.cc und in den sozialen Medien mitteilen. Wir freuen uns auf Ihr Feedback. Viel Spaß!

C. J. Abate (Content Director, Elektor)

INHALT

Q&A mit Martino Facchin

(Leitender Firmware-Ingenieur, Arduino)



Sie können Doom auf einem Portenta-Board laufen lassen!

Doom spielen auf einer Portenta

Retro-Spiele mit einem Arduino

Von David Cuartielles (Arduino)

Du kannst Doom auf einem Arduino Portenta H7 spielen. Neugierig, wie das geht? Willst du wissen, warum die Arduino-Ingenieure das Spiel überhaupt auf dem Gerät haben laufen lassen? Martino Facchin, Leiter des Firmware-Teams von Arduino, hat die Antworten.

Doom ist mit über 3,5 Millionen verkauften Exemplaren wahrscheinlich das meistverkaufte Spiel der Geschichte. Mit einem Verkaufspreis von 50 US-Dollar wurden die Doom-Entwickler, Gründer eines kleinen Unternehmens namens id Software, über Nacht zu Millionären. Als Doom 1993 veröffentlicht wurde, hatte id Software bereits ein Jahr zuvor ein anderes bekanntes Spiel auf den Markt gebracht, Wolfenstein 3D. Doom ist ein Ego-Shooter-Spiel, bei dem der Spieler in einem 3D-Raum navigieren und verschiedene Feinde mit unterschiedlichen Waffen und Munition bekämpfen muss, die auch auf dem gesamten Schlachtfeld des Spiels zu finden sind. Doom wurde auf alle Betriebssysteme portiert, läuft aber auch auf mehreren Systemen in Maschinencode. Der Quellcode ist offen, unter GPL-Lizenz. Obwohl es nicht trivial ist, es zu kompilieren, haben wir gesehen, dass Versionen von Doom auf sehr kleinen Computern und auch innerhalb anderer Programme laufen. Irgendwann hat Microsoft Office Excel 95 eine Hommage an Doom in Form eines spielbaren Ostereis gemacht, das auf der Credits-Seite der Software zu finden ist.

Das Spiel ist zu einem Mittel geworden, um die Leistung kleiner Computer zu testen und Hackerfähigkeiten zu demonstrieren. Vor ein paar Monaten, auf der DEF CON 22 in Las Vegas, führte der Hacker, der auf Twitter und GitHub als @sickcodes bekannt ist, Doom auf einem John Deere-Traktor vor, der so modifiziert war, dass er landwirtschaftliche Grafiken enthielt. Arduino ist da keine Ausnahme. Und als der erste Prototyp des Portenta H7 - unser im Jahr 2018 leistungsstärkstes Board - herauskam, haben wir Doom verwendet, um die technischen Möglichkeiten des Boards zu testen. Vor kurzem habe ich Martino Facchin, den Leiter des

Arduino-Firmware-Teams, eingeladen, um uns mehr über diese Geschichte und die Vorgehensweise zu erzählen.

David Cuartielles: Lass uns über den Arduino Portenta H7 sprechen, auf dem Doom läuft. Ich habe eine kleine Zusammenfassung der Geschichte von Doom vorbereitet, wie die Jungs von id Software das Spiel entwickelten, wie es sich wie verrückt verkaufte und wie die Entwickler zu Millionären wurden.
Martino Facchin: Und dann haben sie den Quellcode veröffentlicht – das ist das Wichtigste.

Cuartielles: Ganz genau! Sie haben den Quellcode veröffentlicht, aber unter welcher Lizenz haben sie ihn veröffentlicht?
Facchin: Ich muss nachsehen, aber ich glaube, die Doom Source Code Licence ist mit der GPL kompatibel. Das Wichtigste ist, dass nur die Engine Open Source ist. Die Assets sind geschützt, und tatsächlich kann man nur die Shareware-Version von Doom spielen, also nicht wirklich das volle Spiel, es sei denn, man hat es gekauft.

Eine Anmerkung von David
Ich habe das überprüft, weil ich mir sicher sein wollte. 1997 veröffentlichte id Software Doom unter der oben genannten Lizenz, die den Quellcode für Bildungszwecke freigab. Nach einem Unfall, der den Entwicklern von glDoom passierte und der dazu führte, dass die Welt keine Kopie der OpenGL-Portierung von Doom hatte, weil die Doom-Lizenz eine Nichtweitergabeklausel enthielt, stimmte id Software zu, die Lizenz in GPL zu ändern.

Cuartielles: Das Tolle an diesem Spiel ist, dass die Leute ihre eigenen Mods gemacht haben. Ich erinnere mich an eine Version von Wolfenstein 3D mit Star Wars-Charakteren.
Facchin: Daran kann ich mich nicht erinnern.

Cuartielles: Ich bin älter. Ich erinnere mich für uns beide.
Facchin: Nun, ich erinnere mich, dass ich als Kind Wolfenstein 3D gespielt habe. Aber wir hatten keine Internetverbindung, also konnten wir nicht all diese Goodies von der Modder-Community bekommen.

Cuartielles: Bevor wir fortfahren, möchte ich Sie bitten, sich zuerst vorzustellen.
Facchin: [lacht] Martino Facchin, Firmware-Ingenieur bei Arduino.

Cuartielles: Was ist Ihre Rolle bei Arduino?
Facchin: Ich bin der Leiter der Firmware-Abteilung – der wichtigste Ansprechpartner, wenn Sie Hilfe bei der Firmware benötigen. Jetzt habe ich ein wunderbares Team von Kollegen, die mir dabei helfen, denn ich habe dieses Team ganz allein gegründet. Das Team wächst ständig weiter. Wir versuchen auch, die Community mit uns wachsen zu lassen, indem wir alle auf unsere Arbeit aufmerksam machen.

Cuartielles: Abgesehen davon, dass Sie Doom auf einem Portenta zum Laufen gebracht haben (worüber wir später noch sprechen werden), was ist es, was Sie bei Arduino gemacht haben, worauf Sie am meisten stolz sind?
Facchin: Ich würde sagen, dass es das Pluggable-USB-Framework ist. Als ich gerade zu Arduino kam und sechs Monate dort war, hatten wir ein großes Problem mit Leuten, die dem USB-Anschluss des Arduino Due und des – damals noch kommenden – Arduino Zero mehrere Funktionen hinzufügen wollten. Jedesmal, wenn man einen Arduino Leonardo an einen Computer anschloss, wurden die Tastaturtreiber, die Maustreiber und so weiter aufgerufen, auch wenn man sie nicht benutzte. Wir mussten also spontan einen USB-Descriptor entwickeln, der den Benutzern nur die Dinge anzeigt, die sie gerade benutzen wollen. Gleichzeitig ließen wir auch andere Dinge zu, die die Leute nutzen wollten, wie USB-MIDI und dergleichen. Für mich war das eine große Entwicklung. Ich war damals noch ziemlich jung und musste mithilfe von Matthijs Kooijman (lesen Sie mehr über seine Arbeit unter www.stderr.nl) und Paul Stoffregen (Erfinder von Teensy) mit der Community interagieren, um die bestmögliche Strategie zu finden. Und es hat funktioniert. Ab und zu kommen Leute und sagen: „Ich habe die MIDI-Bibliothek für dieses oder jenes verwendet“, oder es gibt sogar einen Entwickler, der jetzt alle Layouts für alle internationalen Tastaturen auf der Grundlage dieses Codes erstellt. Darauf bin ich sehr stolz.

Cuartielles: Sie sagten, dass Sie „damals noch ziemlich jung“ waren. Wie lange arbeiten Sie schon für Arduino?
Facchin: Seit sechseinhalb Jahren. Ich arbeite im Büro in Turin.

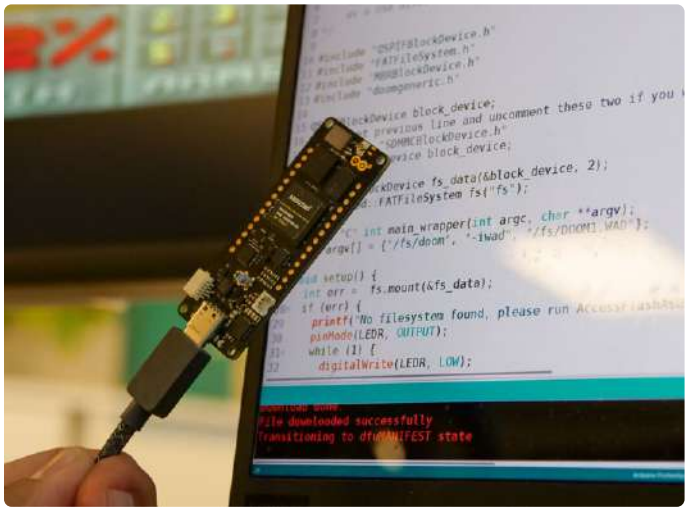
Cuartielles: Das ist eine lange Zeit. Und wie viele Personen sind am Firmware-Team beteiligt?
Facchin: Sechs Personen. Das mag viel erscheinen, aber im Firmware-Team warten wir alle Produkte, während wir auch andere Aktivitäten wie die Zertifizierung durchführen. Andererseits haben wir die Firmware von der Tooling-Abteilung getrennt. Das ist ein anderes Team, das sich mit der Arduino-CLI und anderen Teilen der High-End-Software beschäftigt.

Die offensichtliche Wahl war, Doom zu portieren und zu versuchen, alle Funktionen zu instrumentieren, die für den Betrieb erforderlich waren.

Cuartielles: Und alle Entwickler arbeiten mit Github, richtig?
Facchin: Ja, am Ende des Entwicklungsprozesses ist alles als Open Source verfügbar.

Cuartielles: Großartig. Lassen Sie uns noch einmal über Doom sprechen. Wir wissen, dass es ein sehr erfolgreiches Spiel war, das über 3,5 Millionen Mal verkauft wurde und seine Entwickler stinkreich machte. Es war der erste Bestseller unter den Ego-Shootern (FPS). Der Code wurde als Open Source veröffentlicht und auf alle möglichen Geräte portiert.
Facchin: Telefone, Taschenrechner, alle Betriebssysteme.

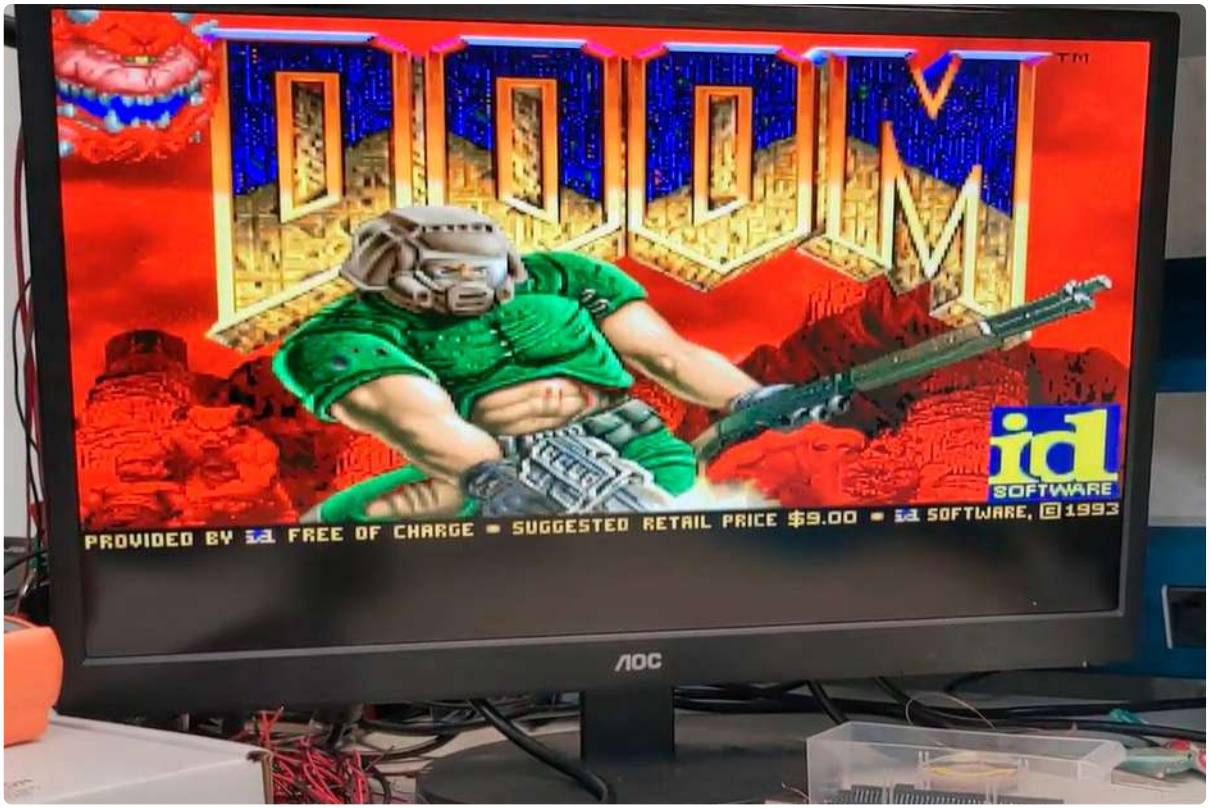
Cuartielles: Und der Portenta H7 von Arduino, ein Dual-Core-Prozessor-Board, das für industrielle Umgebungen gedacht ist. Wer hatte die Idee, Doom auf dem Portenta H7 laufen zu lassen?
Facchin: Es war eher so, dass wir gerade den ersten Prototyp des Portenta H7 bekommen hatten, ein schönes Board mit vielen Chips, und noch nicht ganz fertig. Wir hatten diesen Chip von Analogix, der typischerweise in anderen Geräten zu finden ist und der MIPI-Signale in DisplayPort-Signale umwandelt, was die Ansteuerung externer Monitore ermöglicht. Wir hatten keine Erfahrung mit den Subsystemen des Chips, die dies machen sollten, und die vorhandenen Beispiele waren auch nicht hilfreich. Zunächst gelang es uns, einige gelbe Kästchen auf dem Bildschirm darzustellen, dann das Arduino-Logo mit ein paar Artefakten, aber es war alles andere als perfekt. Wir verstanden nicht wirklich, warum die Dinge nicht wie erwartet funktionierten, also entschied ich mich für etwas mit bewegten Bildern, die ich leicht erkennen konnte.
Die naheliegende Wahl war, Doom zu portieren und zu versuchen, alle Funktionen zu instrumentieren, die nötig waren, um es zum Laufen zu bringen. Es war ein sehr geringer Aufwand, um es auf einem Gerät ohne Betriebssystem zum Laufen zu bringen. Es ist nicht wirklich der Fork von Doom, den man spielen würde, aber es ist sehr einfach zu portieren. Man muss nur sechs oder sieben Funktionen ändern, um es an die eigene Hardware anzupassen, und schon kann es losgehen. Zu Beginn hatten wir keinen Code, um das interne RAM oder den externen Speicher zu betreiben. Ich musste dies erst zum Laufen bringen und dann den Simulator starten. Am Anfang haben wir gar kein Video zum Laufen gebracht. Ich musste einen Bildspeicher vorbereiten, etwas zaubern und eine ordentliche Ausgabe über USB-C hinbekommen.



Es ist Aufgabe des Lesers, das Projekt zu übernehmen und Steuerungen hinzuzufügen, um das Spiel spielbar zu machen.

Wie Sie Doom auf Ihrem Portenta H7 ausführen

1. Laden Sie die neueste Version der Arduino-IDE herunter. Wir empfehlen Arduino 2.0 oder eine neuere Version.
2. Laden Sie den Portenta-Core über den Boardmanager herunter.
3. Wählen Sie den M7-Kern für Ihr Portenta H7-Board. Alle folgenden Schritte müssen auf diesem Kern ausgeführt werden.
4. Vergewissern Sie sich, dass das IDE den Anschluss identifiziert hat, an dem die Karte angeschlossen ist.
5. Es gibt ein Beispiel unter *Datei → Beispiele → Doom*, wo Sie die grundlegenden Anweisungen sehen können. Bevor Sie es installieren, müssen Sie einige Beispiele auf Ihrem Board ausführen.
6. [Optional] Aktualisieren Sie Ihren Portenta-H7-Bootloader auf die neueste Version mit *Datei → Beispiele → STM32H747_System / STM32H747_manageBootloader*.
7. Formatieren Sie den externen Flash *Datei → Beispiele → STM32H747_System / QSPIFormat*. Nach der Installation müssen Sie das serielle Terminal öffnen und den darin enthaltenen Anweisungen folgen.
8. Verwandeln Sie Ihr Board mit *Datei → Beispiele → USB As Mass Storage → AccessFlashAsUSBDisk* in einen Massenspeicher, als wäre es ein USB-Laufwerk.
9. Öffnen Sie den seriellen Monitor und wählen Sie, wie die Formatierung der Karte ablaufen soll. Sobald dies geschehen ist, sollte Ihr Computer zwei neue externe Laufwerke registrieren, die an ihn angeschlossen sind.
10. Laden Sie *DOOM1.WAD* von der DoomWiki-Seite herunter: *https://doomwiki.org/wiki/DOOM1.WAD* und kopieren Sie es in die größte Partition des virtuellen Portenta-Laufwerks.
11. Gehen Sie zurück zu dem *Doom.ino*-Beispiel, das wir in Schritt 5 gesehen haben, und flashen Sie es auf Ihr Board. Denken Sie daran: immer auf dem M7-Kern. Wenn Sie Probleme hatten, den Programmierport zu sehen, doppelklicken Sie einfach den Reset-Knopf auf Ihrem Portenta vor dem Hochladen, so dass der serielle Port richtig erkannt wird.
12. Trennen Sie die Portenta H7 von Ihrem Computer und schließen Sie sie an einen USB-C-Hub an, als wäre das Board ein Laptop. Der Hub muss mit der externen Stromversorgung und einem HDMI-Kabel verbunden sein, um das Videosignal an einen Computermonitor zu senden.



Doom ist ein Klassiker!

Cuartielles: Der Portenta H7 ist ein Dual-Core-Prozessor. Er verfügt über einen Arm Cortex M4 und einen M7. Auf welchem von beiden läuft Doom?
Facchin: Heute verwenden wir den M7, aber damals haben wir ihn auf dem M4 laufen lassen, weil er vom Standpunkt eines Embedded-Programmierers aus viel einfacher war. Er ist mehr wie ein typischer Mikrocontroller, ohne irgendwelche besonderen Eigenschaften. Andererseits hat der M7 diesen Cache, in den man hineinschauen und ihn zum richtigen Zeitpunkt ungültig machen muss, wenn man ein Video erzeugt. Als ich das erste Mal versuchte, Video auf dem M7 laufen zu lassen, war es zwar sehr schnell, aber auch völlig zerstört, und ich konnte nichts auf dem Bildschirm erkennen. Der M4-Kern war dagegen schnell genug (25 Bilder pro Sekunde) und pixelgenau.

Cuartielles: 25 fps ist viel schneller als bei meinem ersten Computer. Aber lassen Sie uns das für den Leser zusammenfassen: Sie hatten Doom auf dem langsameren der beiden Prozessoren laufen, auf einem Board, das Bluetooth- und WLAN-Konnektivität hat. Video wird über USB-C gesendet. Dort kann man einen Hub, eine Maus, eine Tastatur, was auch immer anschließen. Welche Bedienelemente haben Sie dort implementiert?
Facchin: Als wir sahen, dass wir das Video zum Laufen bringen konnten, begannen wir mit LVGL zu arbeiten, einer viel nützlicheren Bibliothek für andere Entwickler, um Anwendungen auf Portenta aufzubauen. LVGL ist vollständig in dem USB-Hub, der Tastatur und der Maus integriert, sodass man alle benötigten Schnittstellen für den professionellen Kontext bauen kann, den viele Arduino-Endbenutzer benötigen.

Cuartielles: Wäre es nicht cool, eine professionelle SPS auf der Basis des Portenta H7 zu haben, bei der man auf dem Cortex M4-Kern Doom spielen könnte, während der M7 die ernsthafte Arbeit erledigt?
Facchin: Auf jeden Fall!

Cuartielles: Danke, Martino. Es war toll, die Geschichte von Doom auf dem Arduino Portenta H7 zu hören. Wir werden die grundlegenden Anweisungen, wie man es zum Laufen bringt, mit unserer Community teilen. Es liegt an den Lesern, das Projekt zu übernehmen und Steuerungen hinzuzufügen, um das Spiel spielbar zu machen.

(220542-02)WdH

Über den Autor

David Cuartielles ist Mitbegründer von Arduino. Er hat einen Doktortitel in Interaktionsdesign, einen MSc in Telekommunikationstechnik und lehrt an der Universität Malmö.

Haben Sie Fragen oder Kommentare?

Haben Sie Fragen oder Anmerkungen zu diesem Artikel? Kontaktieren Sie das Team von Elektor unter redaktion@elektor.de.

20% Rabatt
auf das erste Jahr Ihrer Mitgliedschaft

Treten Sie jetzt der Elektor Community bei!

Werden Sie elektor GOLD membership oder elektor GREEN membership Mitglied!

Gold	Green	
✓	✓	Komplettes Webarchiv ab 1970
✓	☐	8x Elektor Doppelheft (Print)
✓	✓	8x Elektor Digital (PDF)
✓	✓	10% Rabatt im Online-Shop und exklusive Angebote
✓	✓	Zugriff auf über 5.000 Gerber-Dateien aus Elektor Labs
✓	✓	Kostenlose Lieferung innerhalb Deutschlands

www.elektormagazine.de/arduino-member

Nutzen Sie den Gutscheincode:

ARDUINO22

elektor

Ausgepackt

Das Elektor-LCR-Meter **mit David Cuartielles**

Termin: 26. Januar 2023

Möchten Sie den LCR-Meter-Bausatz von Elektor mit mir zusammen auspacken? Dann schauen Sie sich den Elektor Lab Talk am 26. Januar 2023 (18:00 Uhr MEZ) an! In der Diskussion spreche ich zusammen mit den Elektor-Ingenieuren Mathias Claussen und Jens Nickel über das LCR-Meter-Kit und beantworte Ihre Fragen zur Arduino-Technologie und dieser Gastausgabe von Elektor. Verpassen Sie nicht den Livestream und stellen Sie ihre Fragen!

(220555-02)RG

Elektor LabTalk

Sehen Sie David live beim Elektor Lab Talk am 26. Januar 2023!

www.elektormagazine.com/labtalk-david

Crashkurs für Arduino-Einsteiger

Erweiterungsboard für den Arduino Nano

Von Wolfgang Trampert (Deutschland)

Elektor bleibt seinem Bildungsauftrag treu: Ein neues Trainingsboard auf Basis eines Arduino Nano eignet sich ideal für einen Einstieg in die Welt der Mikrocontroller. Zusammen mit einem ausgereiften Schnellkurs bietet es Interessierten eine solide Grundlage, um die praktischen Kenntnisse immer weiter wachsen zu lassen.

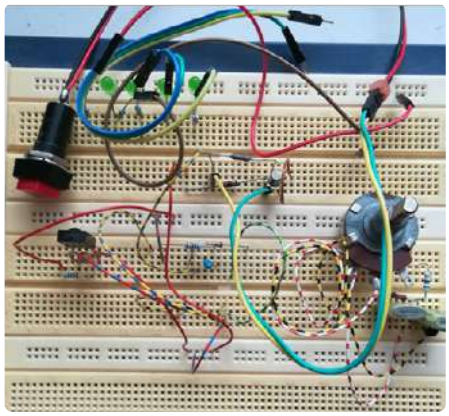


Bild 1: Breadboard-Schaltung für einen Arduino Sketch.

Die Arduino-Philosophie basiert auf einem hardware-nahen Konzept: In den meisten Fällen greift der Arduino-Sketch über die GPIOs des Mikrocontrollers direkt auf Bauelemente wie Schalter, Taster, Potentiometer, Leuchtdioden, LC-Displays, Piezo-Summer, Treibertransistoren oder über die diversen seriellen Schnittstellen wie SPI, I²C oder 1-Draht-Bus auf elektronische Module wie Sensor-, Display- oder Treiber-Boards zu. Wer sich in die Welt der Mikrocontroller- und Arduino-Boards einarbeiten möchte, muss immer wieder neue Übungsschaltungen aufbauen und mit seinem Arduino-Board steuern.

Das Wesentliche bei einem Arduino-Projekt ist jedoch die Entwicklung der Software (Sketch), die Hardware ist nur das Mittel zum Zweck. Man kann diese auf einem Breadboard (Steckbrett) aufbauen (**Bild 1**), allerdings ist das Resultat unübersichtlich und der durch den Aufbau der Schaltung gewonnene Lerneffekt dann nicht besonders groß. Die Zeit, die man für die „Steckerei" aufwenden muss, ist oft bedeutend länger als die Zeit für die Entwicklung des Sketches selbst. Zusätzlich birgt eine solche Schaltungsanordnung eine nicht zu unterschätzende Fehlerquelle. Wie leicht ist ein Bauteil, eine Drahtbrücke oder ein Dupont-Kabel in ein falsches Kontaktloch auf dem Breadboard gesteckt! Die Konsequenz ist eine langwierige Suche nach dem Fehler. Und in dem „Drahtverhau" ist ein Fehler nicht gerade schnell zu entdecken.

Trainingsboard

Aus diesem Grund wurde das Elektor-Arduino-Trainingsboard, auch *MCCAB-Trainingsboard* genannt, entwickelt (**Bild 2**). Es wird von

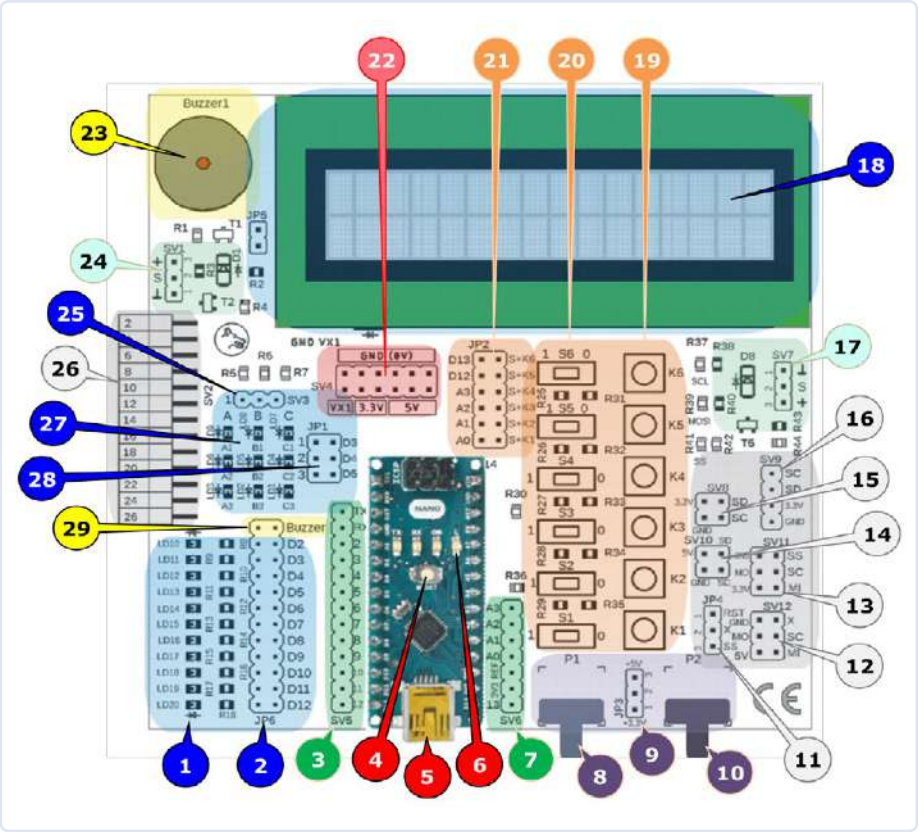


Bild 2: Das MCCAB-Trainingsboard, Rev. 3.3.

Die Bedien- und Anzeigeelemente auf dem MCCAB-Trainingsboard

1	11 × LED (Zustandsanzeige für die Ein- /Ausgänge D2 ... D12)
2	Verbindung der LEDs LD10 ... LD20 mit GPIOs D2 ... D12
3	Mikrocontroller-Ein-/Ausgänge
4	*RESET*-Taster
5	Arduino NANO mit Mini USB-Buchse
6	LED *L*, verbunden mit GPIO D13
7	Mikrocontroller-Ein-/Ausgänge
8	Potentiometer P1
9	Betriebsspannung P1 und P2
10	Potentiometer P2
11	Signal an Pin X von SV12
12	SPI-Interface 5 V (das Signal an Pin X wird über JP4 ausgewählt)
13	SPI-Interface 3.3 V
14	I²C-Interface 5 V
15	I²C-Interface 3.3 V
16	I²C-Interface 3.3 V
17	Schaltausgang für externe Geräte
18	LC-Display mit 2 x 16 Zeichen
19	6 × Tastschalter K1 ... K6
20	6 × Schiebeschalter S1 ... S6
21	Verbindung der Schalter mit den Eingängen des Mikrocontrollers
22	Verteiler für die Betriebsspannungen
23	Piezo-Summer *Buzzer1*
24	Schaltausgang für Geräte
25	Spalten der 3 × 3 LED-Matrix
26	2 × 13 Pins für externe Module
27	3 × 3 LED-Matrix (rot)
28	Verbindung der Reihen der LED-Matrix mit GPIOs D3 ... D5
29	Steckbrücke verbindet *Buzzer1* mit GPIO D9

einem auf das MCCAB-Trainingsboard aufgesteckten Arduino Nano gesteuert und enthält die grundlegenden Bauteile, die im Allgemeinen für die Prototypen neuer Entwicklungen, Laboraufbauten, Test- und Versuchsschaltungen, Projekte und Übungen in Studium und Ausbildung und last but not least im Hobbybereich benötigt werden. Durch Steckbrücken (Jumper) auf den Pfostenleisten des MCCAB Trainingsboards kann der Anwender die für sein jeweiliges Projekt benötigten Bauteile einfach mit den GPIOs des Mikrocontrollers auf dem MCCAB-Trainingsboard verbinden. Fehlerquellen durch falsch gesteckte Verbindungskabel sind dadurch von vornherein eliminiert, ebenso wie die lästige Suche nach den gerade benötigten Bauteilen in der Bastelkiste, denn alle Bauteile befinden sich bereits „on Board".

Auch der Anschluss von auf Breadboards aufgebauten Zusatzschaltungen ist auf einfache Weise über Dupont-Kabel möglich, denn alle GPIOs des Mikrocontrollers auf dem Arduino Nano sind auf dem MCCAB-Trainingsboard an die beiden Pfostenleisten SV5 und SV6 (Pfeil 3 und Pfeil 7 in Bild 2) gelegt. Links in Bild 2 sieht man darüber hinaus die 26-polige Buchsenleiste SV2 (Pfeil 26) zum Anschluss externer Platinen, an die ebenfalls alle wichtigen GPIOs des Mikrocontrollers geführt sind. So können externe Platinen wie zum Beispiel eine Schaltung zur Aufnahme der Kennlinien elektronischer Bauelemente, das Abbild eines Straßenzugs mit Ampelschaltungen oder ein elektronisches Labornetzteil einfach an das MCCAB-Trainingsboard angedockt und von ihm gesteuert werden. Auf dem LC-Display mit 2 x 16 Zeichen (Pfeil 18), das über den I²C-Bus gesteuert wird, lassen sich aus dem laufenden Sketch heraus Ergebnisse und Hinweise anzeigen. Auch eine aus drei Spalten und drei Zeilen bestehende LED-Matrix (Pfeil 27) ist auf dem Board vorhanden.

Das MCCAB-Trainingsboard arbeitet mit einer Betriebsspannung von Vcc = +5 V. Die Stromversorgung erfolgt in der Regel über die USB-Schnittstelle des angeschlossenen PCs, der für die Erstellung der Übungssketche sowieso benötigt wird. Alternativ ist auch eine Stromversorgung durch ein externes Netzteil möglich.

In der schematischen Ansicht des Trainingsboards (Bild 2) sind die Bauteile zusammengehöriger Gruppen in der gleichen Farbe schattiert.

Die Bibliothek *MCCAB_Lib* für das Trainingsboards

Seinen Sketch entwickelt der Anwender wie gewohnt in der Arduino-IDE auf seinem PC und lädt ihn über ein Mini-USB-Kabel in den Mikrocontroller des Arduino Nano auf dem Trainingsboard.

Tabelle 1: Die in der Library MCCAB_Lib enthaltenen Klassen.

Klasse	Verwendungszweck
`KeySwitch`	Prellfreie Abfrage der Schalter S1 ... S6 bzw. der Taster K1 ... K6
`Matrix`	Steuerung der Anzeige der 3 × 3 LED-Matrix
`LED`	Ein / Aus / Blinken der 12 Leuchtdioden LD10 ... LD20 sowie LED
`LedBlock`	Gleichzeitige Ausgabe eines Bitmusters an alle 11 Leuchtdioden *LD10 ... LD20*
`Sound`	Steuerung des Summers Buzzer1 und Erzeugung von Rechtecksignalen

Die GPIOs des Mikrocontrollers lassen sich wie gewohnt mit der Arduino-Funktion `pinMode()` konfigurieren und die Bauteile auf dem Trainingsboard mit `digitalRead()`, `digitalWrite()`, `analogRead()` ... abfragen beziehungsweise steuern. Zur Unterstützung des Anwenders bei der Steuerung der umfangreichen Hardware-Peripherie auf dem MCCAB-Trainingsboard ist aber auch die Library *MCCAB_Lib* [1] verfügbar, die kostenlos heruntergeladen und in den eigenen Sketch eingebunden werden kann. Diese Library erleichtert das Handling der Komponenten auf dem Board deutlich.
Die Bibliothek *MCCAB_Lib* enthält fünf Klassen zur Steuerung der auf dem Trainingsboard befindlichen Schalter, der Leuchtdioden und des Summers, die vom Anwender in seinem Sketch beliebig verwendet werden können. **Tabelle 1** zeigt eine Auflistung der verfügbaren Klassen.
Um das Timing zum Entprellen der Schalter, zur Steuerung des Multiplex-Betriebs der 3x3-LED-Matrix und des Blinkens der LEDs (*LD10 ... LD20* beziehungsweise *L*) sowie zur Erzeugung der Tonfrequenzen für den Summer braucht sich der Anwender nicht zu kümmern. Dies erledigt die Bibliothek automatisch und vom Anwender unbemerkt im Hintergrund des Programmablaufs.

Der kleine Beispiel-Sketch in **Listing 1** zeigt die Verwendung der Bibliothek *MCCAB_Lib.* In Zeile 15 des Sketches wird die Objektvariable `Led` der Klasse `LED` aus der Bibliothek *MCCAB_Lib* deklariert. Der bei der Deklaration der Objekt-Variablen `Led` übergebene Parameter `LED_PIN` ist die in Zeile 13 definierte Konstante für den Pin, an den die LED angeschlossen ist. Der Pin wird bei der Instanziierung automatisch als Ausgang konfiguriert.
Die in Zeile 22 deklarierte Objektvariable `Key` der Klasse `KeySwitch` aus der Library *MCCAB_Lib* überwacht im Hintergrund des Sketches den Zustand des Schalters am Pin *SK4*, der bei der Deklaration als Parameter übergebenen wurde, unterdrückt die Prellimpulse des Tasters K4 beim Öffnen oder Schließen und ruft immer dann die Funktion `switchTurnedOn()` auf, wenn der Taster gedrückt wird. Die in der Funktion `switchTurnedOn()` in Zeile 19 aktivierte Methode `toggle()` der Klasse `LED` aus der Bibliothek *MCCAB_Lib* invertiert den aktuellen Zustand der Leuchtdiode LD10.

Listing 1. Sketch zum rastenden Ein/Ausschalten der LED mit Taster K4

```
/*
* Sketch zum rastenden Ein- / Ausschalten der LED LD10 mit dem Taster K4 unter Verwendung
* von Objektvariablen der Klassen "KeySwitch" und „LED“ der Bibliothek MCCAB_Lib.
* Um den Zustand des Tasters K4 über den GPIO A3 des Mikrocontrollers einlesen zu können,
* muss eine Brücke (Jumper) auf Position S+K4 der Pfostenleiste JP2 auf dem MCCAB
* Trainingsboard gesteckt sein.
* Um die LED LD10 mit dem GPIO des Mikrocontrollers steuern zu können, muss eine Brücke auf
* Position D2 der Pfostenleiste JP6 auf dem MCCAB Trainingsboard gesteckt sein.
*/

#include <MCCAB_Lib.h> // Einbinden der Library MCCAB_Lib in den Sketch

#define LED_PIN  2     // die LED ist an Pin D2 angeschlossen

LED Led(LED_PIN);      // Objekt-Variable

//Funktion, die von der Objekt-Variablen "Key" beim Schließen des Schalters aufgerufen wird
void switchTurnedOn() {
  Led.toggle();        // der Zustand der LED wird umgeschaltet (invertiert)
}

KeySwitch Key(SK4, ACTIVE_HIGH, switchTurnedOn, nullptr);  // Objekt-Variable

void setup() { }   // nichts zu tun
void loop() { }    // nichts zu tun
```

Da die Anschlusspins des Schalters und der LED bei der Deklaration der Objekte automatisch als Eingang oder Ausgang konfiguriert werden, ist in der Funktion `setup()` in Zeile 24 in diesem Sketch nichts weiter zu tun.
Auch die Funktion `loop()` in Zeile 26 enthält keine Anweisungen, denn die einzige Aktion, die in diesem Sketch auszuführen ist, ist das Umschalten des LED-Zustandes beim Drücken des Tasters K4. Diese Aktion wird ereignisgesteuert von der Klasse `KeySwitch` durch den Aufruf der Funktion `switchTurnedOn()` ausgeführt.
In einem umfangreicheren Sketch als diesem wären somit die beiden Funktionen `setup()` und `loop()` durch den Einsatz der Bibliothek *MCCAB_Lib* von der leidigen Überwachung der Peripheriekomponenten entlastet; sie könnten sich den „wirklich wichtigen" Dingen widmen.

Zwölf Projekt-Sketche und 46 Übungen

Für das MCCAB-Trainingsboard gibt es eine ausführliche Bedienungsanleitung, die von der Webseite [1] heruntergeladen werden kann. Außerdem sind das MCCAB-Trainingsboard und die Bibliothek *MCCAB_Lib* in dem im Elektor-Verlag erschienenen Lehrbuch *Mikrocontroller-Praxiskurs für Arduino-Einsteiger* (ISBN 978-3-89576-523-0) detailliert beschrieben.

Das Buch erklärt ausführlich die Grundlagen der Hard- und Software eines Mikrocontrollersystems und führt ein in die Programmiersprache C, in der die Arduino-Sketche geschrieben werden.
Der Schwerpunkt des Buches liegt aber auf den praktischen Übungen, denn der Leser eignet sich die erforderlichen Kenntnisse durch „Learning by Doing" an: In einem umfangreichen Praxisteil mit zwölf Projekt-Sketches und 46 Übungen wird das Gelernte mit vielen Beispielen unterlegt. Die Übungen sind dabei so aufgebaut, dass der Nutzer eine Aufgabenstellung erhält, die er zuerst mit dem MCCAB-Trainingsboard und seinem im Theorieteil des Buches aufgebauten Wissen lösen sollte. Für jede Übung gibt es anschließend eine ausführlich erklärte und kommentierte Musterlösung, die bei Problemen weiterhilft.

(220450-02)RG

Sie haben Fragen oder Kommentare?

Haben Sie technische Fragen oder Anmerkungen zu diesem Artikel? Dann schreiben Sie an die Elektor-Redaktion unter redaktion@elektor.de.

Über den Autor

Wolfgang Trampert beschäftigt sich seit seinem Elektronik-Studium mit dem Bau und der Programmierung von Mikrocontrollern und ihrer Peripherie. Als Inhaber eines Ingenieurbüros hat er im Kundenauftrag von Mikrocontrollern gesteuerte Produkte entwickelt. Er ist als Autor von Fachbüchern und -artikeln tätig und leitet Schulungen zum Thema Mikrocontroller.

Passende Produkte

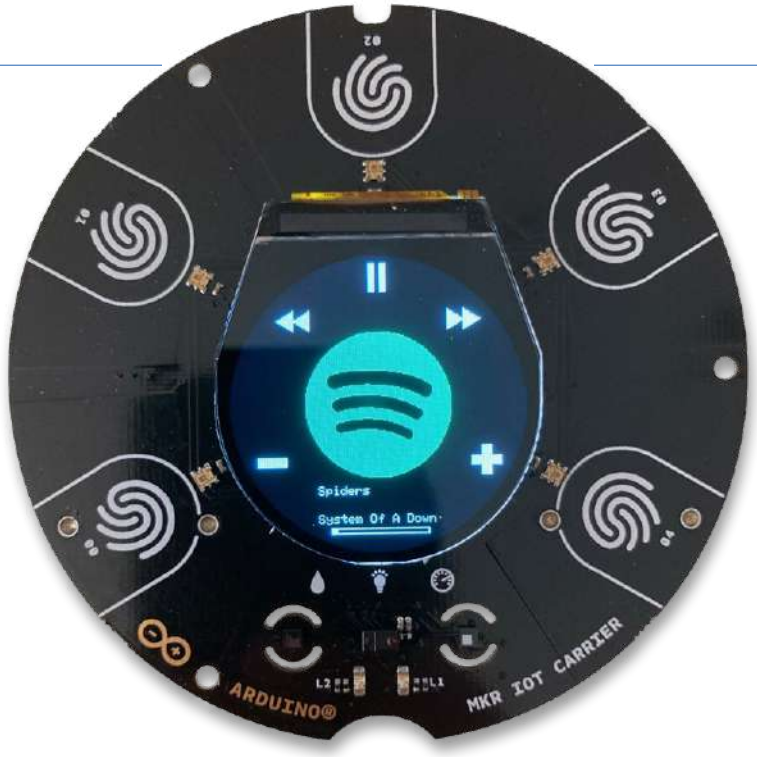
- **MCCAB-Trainingboard (SKU 20295)**
  www.elektor.de/20295
- ***Mikrocontroller-Praxiskurs für Arduino-Einsteiger* (Buch, SKU 20293)**
  www.elektor.de/20293

WEBLINK

[1] Bibliothek MCCAB_Lib: http://www.elektor.de/20295

Ein Controller für Spotify

Das Oplà IoT Kit ist (fast) alles, was Sie brauchen



Von Altuğ Bakan (Türkei)

Das Arduino Oplà IoT-Kit enthält das MKR WiFi 1010 Maker Board und eine Trägerplatine, auf der Relais, ein rundes OLED-Display, kapazitive Touch-Tasten und einige Sensoren integriert sind. Hier beschreiben wir, wie man einen tragbaren Controller für den beliebten Spotify-Musikplayer baut. Natürlich ist eine gewisse Sicherheit erforderlich.

Das *Arduino MKR WiFi 1010 Maker Board* ist dank seiner WLAN-Fähigkeiten ein perfektes Gehirn für Ihr nächstes IoT-Projekt. Noch besser ausgestattet sind Sie mit dem *Arduino Oplà IoT-Kit*, das dieses Maker Board und eine Trägerplatine enthält (**Bild 1**). Auf letzterer sind Relais, ein rundes OLED-Display und kapazitive Berührungs-Tasten integriert. Außerdem enthält der Bausatz einen Feuchtigkeits- und einen PIR-Sensor (**Bild 2**). Projekte wie Haussicherheitsalarme und automatische Pflanzenbewässerung sind damit leicht zu realisieren.

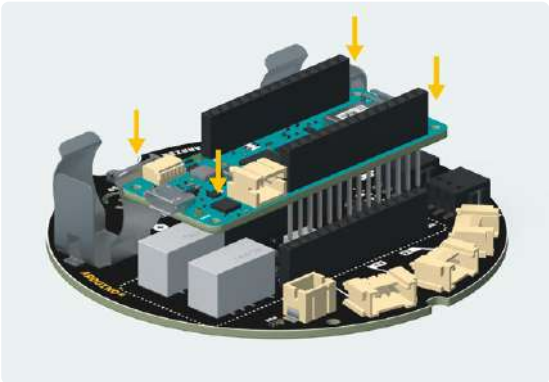


Bild 1. Das MKR WiFi 1010 Maker Board wird auf die Trägerplatine gesteckt, auf der Relais und andere nützliche Peripheriegeräte integriert sind.

Die WLAN-Funktion ermöglicht auch die Steuerung von Programmen, die auf Ihrem PC laufen, sofern diese über eine Netzwerkschnittstelle verfügen. Die Touch-Tasten, das Display, der Batteriehalter und ein Gehäuse machen es einfach, einen tragbaren Controller für verschiedene Arten von PC-Software zu entwickeln - als Ergänzung zu Maus und Tastatur (**Bild 3**).

Ich bin ein Fan des Musik-Players Spotify und habe daher den Oplà-Bausatz verwendet, um meinen eigenen drahtlosen Spotify-Controller zu bauen. Mit den Tasten kann man zum nächsten und vorherigen Lied springen, ein Lied abspielen/anhalten und die Lautstärke erhöhen und verringern. Dazu muss natürlich der Spotify-Player auf dem PC oder Smartphone gestartet sein.

Sichere Kommunikation

Spotify verfügt über eine einfach zu bedienende Programmierschnittstelle zur Steuerung Ihres Spotify-Players über das Netzwerk, für die Sie allerdings die Spotify-Plus-Lizenz benötigen. Natürlich ist eine gewisse Sicherheit erforderlich. Um die auf REST basierende Spotify-Web-API nutzen zu können, müssen Sie sich zunächst mit Ihrem Spotify-Benutzernamen und -Passwort beim Spotify-Accounts-Server authentifizieren. Sobald Sie authentifiziert sind, muss Ihre Software eine Client-ID und ein Client-Secret senden. Der Spotify-Server gibt ein Zugriffstoken zurück, das Sie bei jedem Aufruf der Web-API senden müssen, um Ihren Spotify-Player zu steuern. Dieser zweis-

Bild 2. Das Arduino Oplà IoT-Kit.

Bild 3. Der Batteriehalter macht das Oplà IoT-Kit mobil..

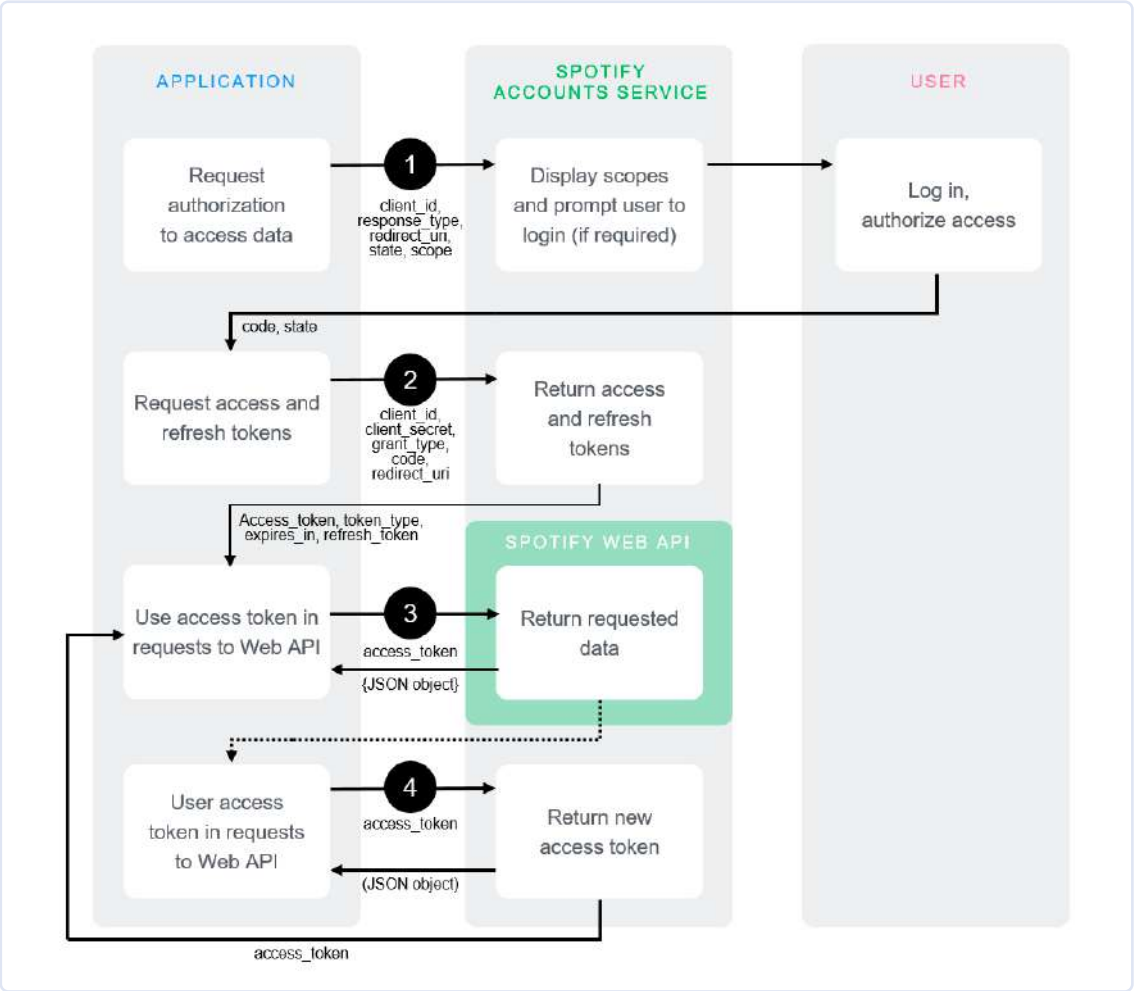


Bild 4. Der mehrstufige Authentifizierungsablauf basiert auf dem beliebten OAuth 2.0-Verfahren.

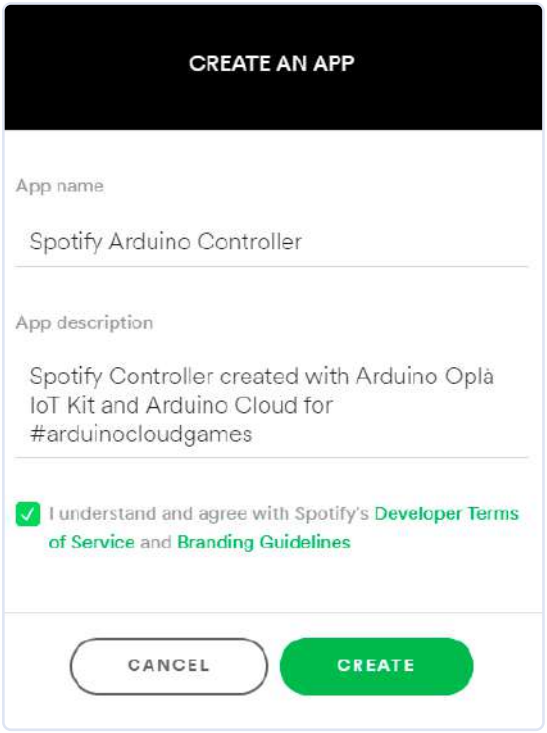


Bild 5. Sie müssen eine „App" machen, um …

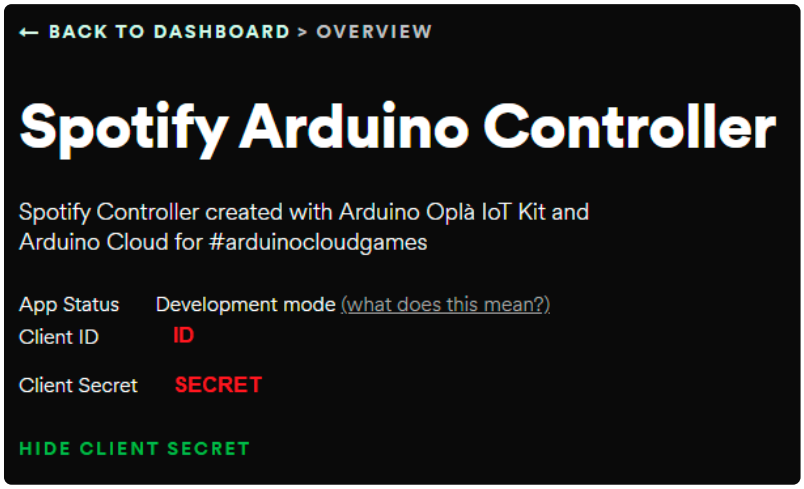


Bild 6. …Ihre Client-ID und Ihr Client-Secret zu bekommen

tufige Authentifizierungsablauf basiert auf dem beliebten *OAuth2*-Verfahren (siehe **Bild 4**).

Wie erhalten Sie Ihre Client-ID und Secret? Verwenden Sie einfach den Spotify App Builder [1], mit dem Sie Ihre eigene PC-Software oder mobile App zur Steuerung von Spotify entwerfen können (**Bild 5**). Das tun wir hier jedoch nicht; wir wollen nur die Anmeldedaten (siehe **Bild 6**). Client ID und Secret müssen auf unserem Arduino MKR Board gespeichert werden. Natürlich könnte man dies im Arduino-Sketch fest einprogrammieren, aber es gibt einen bequemeren und sichereren Weg, dies zu tun. Der Arduino-Web-Editor [2] bietet eine Registerkarte namens *Secrets*, in der Sie Umgebungsvariablen festlegen können, die später in Ihrem Code verwendet werden (**Bild 7**). Geben Sie einfach die Spotify-Client-ID und Secret sowie den Namen und das Passwort Ihres WLANs in die Felder der Registerkarte ein. Wenn Sie die Software kompilieren und auf den Controller hochladen, werden Ihre individuellen geheimen Werte ebenfalls hochgeladen, damit sie vom Projektcode verwendet werden können. In Ihrem Sketch müssen Sie die Strings, die sensible Daten enthalten, durch einen `SECRET_xxx` Ausdruck ersetzen - also zum Beispiel: `SECRET_SPOTIFY_CLIENT`.

Authentifizierung

Um den OAuth2-Flow zu starten, müssen Sie sich bei Spotify authentifizieren. Wenn der hier beschriebene Spotify-Controller gestartet wird, loggt er sich in das angegebene (heimische) WLAN ein und zeigt die IP-Adresse, die er vom Router erhalten hat, auf dem OLED-Display an. Ich wollte dem Benutzer die Möglichkeit geben, sich möglichst einfach bei Spotify zu authentifizieren, also habe ich den folgenden Ansatz entwickelt.

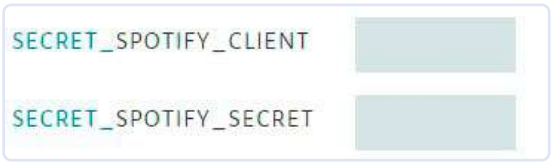


*Bild 7. Geben Sie alle privaten Werte auf der Registerkarte „**Secrets**" des Arduino-Web-Editors ein, bevor Sie den Code kompilieren und hochladen.*

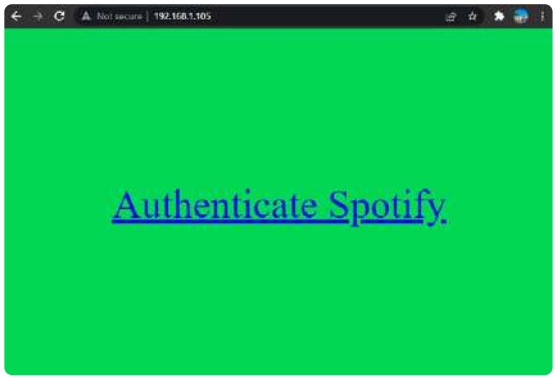


Bild 8. Vom Controller angebotene Webseite zum Einloggen bei Spotify.

Der Arduino Controller generiert eine kleine Webseite, die in einem Webbrowser angezeigt wird, wenn man dort die IP-Adresse des Controllers eingibt (**Bild 8**). Diese kleine Webseite enthält einen Weblink. (Siehe **Listing 1**, um zu sehen, wie die Webseite im Arduino-Code generiert wird). Wenn Sie darauf klicken, wechselt der Browser zur Authentifizierungsseite von Spotify, wo Sie sich einfach anmelden können. Sie werden dann gefragt, ob Sie dem Controller die Erlaubnis geben, Spotify zu steuern (**Bild 9**).

Bitte beachten Sie: Damit dies alles funktioniert, müssen Sie auch die IP-Adresse des Controllers als *Redirect URI* in den Spotify-App-Editor eingeben (**Bild 10)**.
Von nun an kann der Arduino-Controller das API-Zugangs-Token erhalten, indem er die Client-ID und das Secret an Spotify sendet (**Listing 2**). Das Access-Token muss während des Betriebs regelmäßig aktualisiert werden. Dies geschieht ebenfalls über eine Funktion im Sketch (**Listing 3**), die alle 3000 Sekunden aufgerufen wird.

Listing 1: Webseite zur Authentifizierung bei Spotify, angeboten vom Spotify-Controller.

```
String webpage = "<!DOCTYPE html>\n";
webpage += "<html><body>";
webpage += getStyle();
webpage += "<a href=\"https://accounts.spotify.com/authorize?client_id=";
webpage += SPOTIFY_CLIENT;
webpage += "&response_type=code&redirect_uri=http://";
webpage += ip_address;
webpage += "/redirect/&scope=user-read-playback-state user-modify-playback-state\">Authenticate Spotify</a>\n";
webpage += "</body></html>";
wifiClient.print(webpage);
```

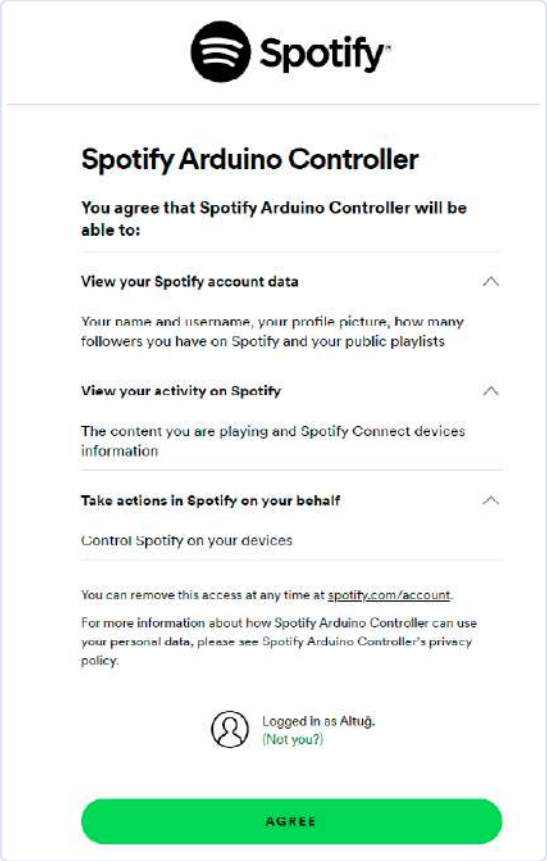


Bild 9. Wenn Sie bei Spotify angemeldet sind, müssen Sie dem Controller die Erlaubnis erteilen, in Ihrem Namen zu handeln.

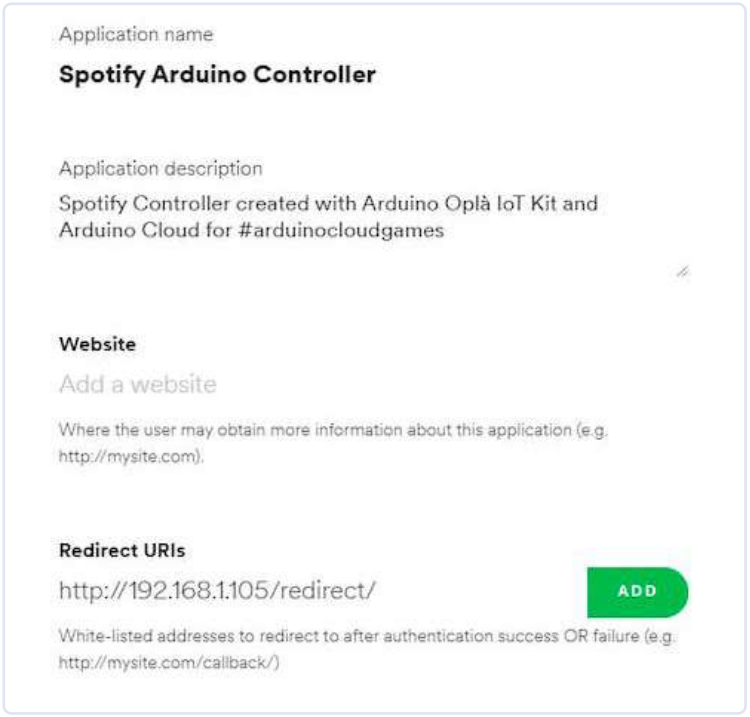


Bild 10. Redirect URI: Die Adresse des Controllers in Ihrem Heimnetzwerk.

Listing 2: Funktion zum Abrufen des Tokens von Spotify für die weitere Verwendung der API.

```
// Get the user authorization token
 bool getAccessToken(String userCode) {
   String postData = "grant_type=authorization_code&code=" + userCode + "&redirect_uri="
                     "http://" + ip_address + "/redirect/";
   authClient.beginRequest();
   authClient.post("/api/token");
   authClient.sendHeader("Content-Type", "application/x-www-form-urlencoded");
   authClient.sendHeader("Content-Length", postData.length());
   authClient.sendBasicAuth(SPOTIFY_CLIENT, SPOTIFY_SECRET);
       // send the client id and secret for authentication
   authClient.beginBody();
   authClient.print(postData);
   authClient.endRequest();

  // If successful
   if (authClient.responseStatusCode() == 200) {
     lastTokenTime = millis();
     DynamicJsonDocument json(512);
     deserializeJson(json, authClient.responseBody());
     accessToken = json["access_token"].as<String>();
     refreshToken = json["refresh_token"].as<String>();
     return true;
   }
   return false;
 }
```

Listing 3: Funktion zur Aktualisierung des Tokens.

```
// Refresh the user authentication token
 void refreshAccessToken() {
   String postData = "grant_type=refresh_token&refresh_token=" + refreshToken;
   authClient.beginRequest();
   authClient.post("/api/token");
   authClient.sendHeader("Content-Type", "application/x-www-form-urlencoded");
   authClient.sendHeader("Content-Length", postData.length());
   authClient.sendBasicAuth(SPOTIFY_CLIENT, SPOTIFY_SECRET);
       // send the client id and secret for authentication
   authClient.beginBody();
   authClient.print(postData);
   authClient.endRequest();

  // If successful
   if (authClient.responseStatusCode() == 200) {
     lastTokenTime = millis();
     DynamicJsonDocument json(256);
     deserializeJson(json, authClient.responseBody());
     accessToken = json["access_token"].as<String>();
   }
 }
```

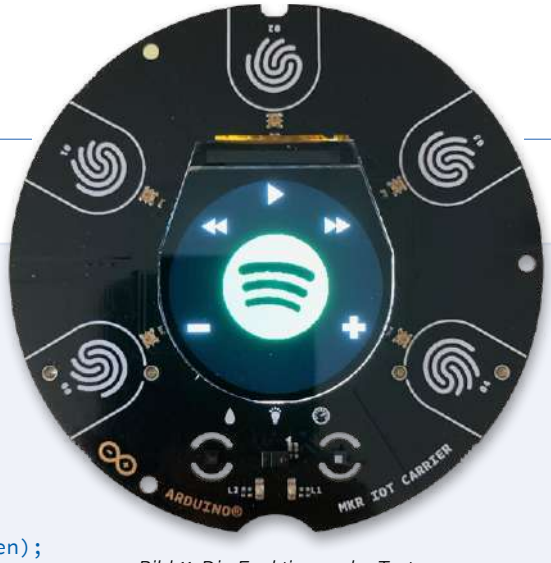


Bild 11. Die Funktionen der Tasten werden auf dem Display angezeigt.

Listing 4: Beispiel für die Verwendung der API (nächster und vorheriger Song).

```
// Skip a song towards a given direction
 void skipSong(String direction) {
   apiClient.beginRequest();
   apiClient.post("/v1/me/player/" + direction);
   apiClient.sendHeader("Content-Length", 0);
   apiClient.sendHeader("Authorization", "Bearer " + accessToken);
   apiClient.endRequest();
 }
```

Betrieb

Der Rest des Codes ist weniger komplex. Das Gerät zeigt das Spotify-Logo und die Funktion der Tasten auf dem OLED-Display an (**Bild 11**). Berührt der Benutzer eine Taste, wird die entsprechende API-Funktion aufgerufen. In **Listing 4** ist zu sehen, wie dies für das Überspringen eines Liedes zum vorherigen oder nächsten gemacht wird.

Es gibt auch eine Funktion im Code, die den Status des Players von der Spotify-API abfragt. Die Antwort ist ein JSON-String. Ich verwende die *ArduinoJson.h*-Bibliothek und einige meiner eigenen Funktionen, um JSON-Strings einfacher zu verarbeiten.

Um den Status der Tasten abzurufen, die LEDs zu steuern und Grafiken auf dem OLED anzuzeigen, verwende ich die Bibliothek `Arduino_MKRIoTCarrier.h`. Studieren Sie meinen Code, um Ideen für eigene Projekte mit dem Oplà Kit zu entwickeln. Meine Software kann unter [3] heruntergeladen werden.

Cloud-Verbindung

Ich habe auch eine Verbindung zur Arduino-Cloud hergestellt und ein Dashboard erstellt, das den aktuellen Titel und den Namen des Interpreten neben der Lautstärke des Geräts anzeigt (**Bild 12**). Natürlich können Sie Ihr eigenes persönliches Dashboard mit den gewünschten Daten erstellen.

Mein Projekt hat den 3. Platz bei den Arduino Cloud Games 2022 gewonnen!

(220407-02)WdH

Bild 12. Der aktuelle Titel und der Name des Interpreten werden an die Arduino Cloud gesendet, wo sie auf dem persönlichen Dashboard angezeigt werden.

Haben Sie Fragen oder Kommentare?

Bei technischen Fragen wenden Sie sich bitte per E-Mail an den Autor unter mail@alt.ug oder an die Elektor-Redaktion unter redaktion@elektor.de.

Über den Autor

Altuğ Bakan arbeitet als Elektronikingenieur, hauptsächlich mit eingebetteten Systemen. Er liebt es, Arduino in seiner Arbeit für ein schnelles Prototyping und Benutzerfreundlichkeit zu verwenden. Seine Lieblingsthemen im Bereich Elektronik sind Bare-Metal-Embedded-Programmierung und Internet of Things (IoT).

Passendes Produkt

› **Arduino Oplà IoT Kit**
www.elektormagazine.de/arduino-opla-iot-kit

WEBLINKS

[1] Spotify App Builder: https://developer.spotify.com/dashboard/
[2] Arduino Web Editor: https://create.arduino.cc/editor
[3] Dieses Projekt auf create.arduino.cc: https://create.arduino.cc/projecthub/Altug/opla-spotify-controller-6e7bc4

Skalierbare, sichere Anwendungen **erstellen, in Betrieb nehmen** und **pflegen**

Arduino Portenta X8 mit dem i.MX 8M Mini-Anwendungsprozessor von NXP und dem EdgeLock® Secure Element SE050

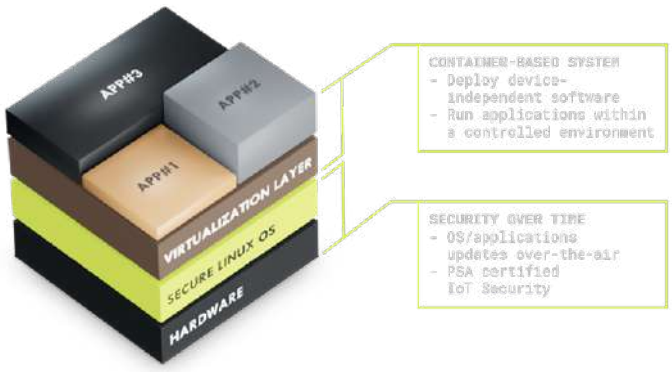
Ein Beitrag von NXP Semiconductors

Ein IoT-Gerät auf den Markt zu bringen ist mit erheblichem Design- und Entwicklungsaufwand verbunden - mit Skalierbarkeitsproblemen, Sicherheitsanforderungen und Gerätebeschränkungen an allen Ecken und Enden. Das Hinzufügen von Intelligenz macht es noch komplizierter. Daher ist die Auswahl der richtigen Entwicklungshard- und -software entscheidend, um sichere Edge-Produkte schneller auf den Markt zu bringen. In diesem Artikel wird die Arduino-Plattform Portenta X8 vorgestellt, ein industrietaugliches, sicheres System on Module (SoM), das auf dem Applikationsprozessor i.MX 8M Mini von NXP und einem integrierten Hardware-Sicherheitselement EdgeLock®-SE050 basiert. Diese PSA-zertifizierte Plattform ist auch *Arm® SystemReady IR* für Linux für garantierte Sicherheit auf Embedded-Arm-SoCs zertifiziert.

Arduino Portenta X8 ist ein leistungsfähiges, industrietaugliches System auf einem Modul mit vorinstalliertem Linux®-Betriebssystem, das dank seiner modularen Container-Architektur in der Lage ist, geräteunabhängige Software auszuführen. Es bietet zwei Ansätze: Flexibilität bei der Nutzung von Linux kombiniert mit Echtzeitanwendungen durch die Arduino-Umgebung. Die integrierte WLAN/Bluetooth®-Low-Energy-Konnektivität ermöglicht die Fernaktualisierung von Betriebssystemen und Anwendungen, wobei die Linux-Kernel-Umgebung stets auf dem höchsten Leistungsniveau gehalten wird.

Sicherheit auf dem neuesten Stand der Technik

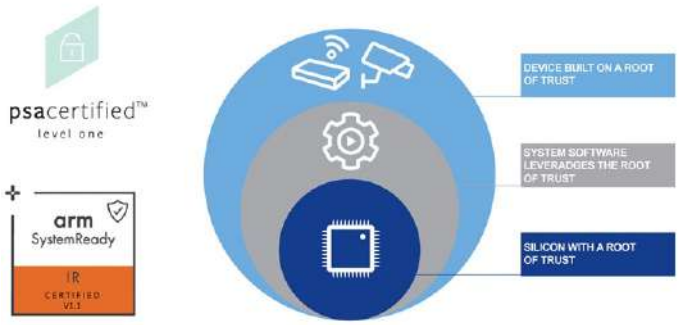
Das containerbasierte System vereint verschiedene Sicherheitsebenen, angefangen bei der Hardware-Ebene, die das Secure Element von NXP enthält. Es nutzt die Cloud-basierte DevOps-Plattform von Foundries.io [1], um die Art und Weise, wie Embedded-Linux-Lösungen entwickelt, getestet, bereitgestellt und gewartet werden, neu zu definieren. Der Portenta X8 enthält das anpassbare Open-Source-Betriebssystem Linux-MicroPlatform, das mit den besten Industriepraktiken für End-to-End-Sicherheit, inkrementellen OTA-Updates und Flottenmanagement entwickelt wurde.



Portenta X8 Container und Sicherheit.

Die Virtualisierungsschicht ermöglicht es Anwendern, geräteunabhängige Software in einer kontrollierten Umgebung auszuführen. Sie können ihre eigenen Container mit Docker erstellen und vorgefertigte Images vom Docker Hub oder anderen öffentlichen Registries herunterladen, um eine maßgeschneiderte Anwendung zu erstellen. Wenn ein Entwickler in die Welt der eingebetteten Systeme einsteigen möchte, kann er dies ganz einfach tun, indem er seine Anwendung erstellt, sie in einem Container überführt, sie auf dem Board installiert und sie sofort testet. Durch die Kombination von Linux-Fähigkeiten und Arduino-Erfahrung ergeben sich zahlreiche Möglichkeiten.
Portenta X8 hat die PSA-Zertifizierung erhalten. Das NXP EdgeLock SE050 Hardware-Sicherheitselement bietet Schlüsselgenerierung, beschleunigte Kryptooperationen und sichere Speicherung. X8 erhielt außerdem die Arm® SystemReady-Zertifizierung [2] und integrierte Parsec-Services, was es zu einem der ersten Cassini-Produkte oder Cloud-Native-Edge-Geräte macht, die für Entwickler auf dem Markt verfügbar sind. Es läuft nahtlos auf Fedora IoT, Fedora Server, Debian und Linux-microPlatform. Der Portenta X8 ermöglicht die Migration von Cloud-nativen Workloads von der Cloud zur Edge und trägt zu

einer Cloud-nativen Entwicklererfahrung im vielfältigen und sicheren IoT-Ökosystem von Arm bei.



Architektur der Plattform-Sicherheit.

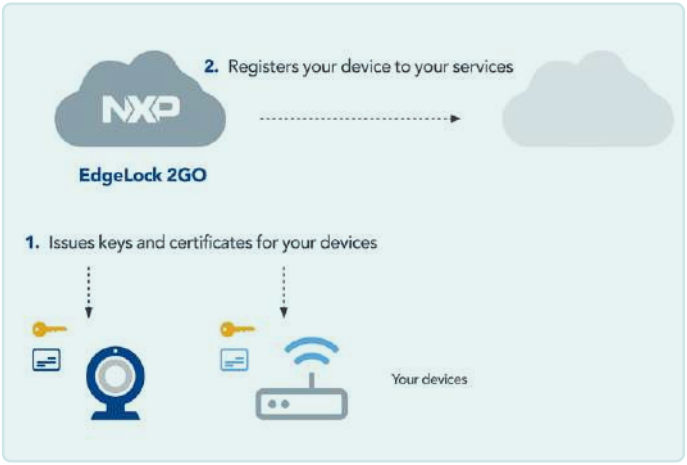
EdgeLock SE050 - Ein Anker des Vertrauens für das IoT

Der EdgeLock SE050 [3] von NXP ist eine diskrete und fälschungssichere Sicherheitshardware zum Schutz der Identität eines Geräts, einschließlich kryptografischer Schlüssel und Zertifikate. Es handelt sich um ein eigenständiges, eingebettetes Sicherheitselement, das über die I²C-Schnittstelle mit dem Hauptprozessor verbunden ist. Der EdgeLock SE050 ist nach *Common Criteria EAL 6+* für die Hardware und das Betriebssystem zertifiziert. Dieses sofort gebrauchsfertige Sicherheitselement für IoT-Geräte bietet eine Vertrauensbasis auf IC-Ebene und liefert echte End-to-End-Sicherheit – von der Edge bis zur Cloud - ohne die Notwendigkeit, Sicherheitscode zu implementieren oder kritische Schlüssel und Berechtigungsnachweise zu verwalten.

Quelle des Vertrauens auf Silizium: Das EdgeLock® SE050 Secure Element.

EdgeLock SE050 als gebrauchsfertige Lösung mit mehreren vorimplementierten kryptografischen Algorithmen und Protokollen sowie einem kompletten Produktunterstützungspaket vereinfacht die Entwicklung und die Markteinführungszeit. Zusätzlich zu den Bibliotheken für verschiedene MCUs und MPUs bietet das Support-Paket auch die Integration mit vielen gängigen Betriebssystemen wie Linux, RTOS und Android.

Entwickler von IoT-Geräten stehen vor zwei großen Herausforderungen, wenn sie Geräte mit der Cloud verbinden wollen: die Bereitstellung der Geräteidentität und die Verwaltung der Geräteidentitäten nach der Freigabe für das Feld. Ersteres bezieht sich auf die Installation von Schlüsseln und Zertifikaten, der zweite Punkt auf die Aktualisierung, das Hinzufügen oder den Entzug von Schlüsseln und Zertifikaten während des gesamten Lebenszyklus des Geräts.
Um Entwickler bei der Bewältigung dieser Herausforderungen zu unterstützen, bietet NXP den *EdgeLock 2GO Managed Service* [4] an. Bei der Plattform handelt es sich um eine speziell entwickelte Hardware- und Servicekombination, die eine siliziumbasierte Vertrauenswürdigkeit (Root of Trust) einrichtet. EdgeLock 2GO stellt die für IoT-Geräte erforderlichen Identitäten aus und installiert die Anmeldedaten sicher in der EdgeLock-SE050-Hardware. Außerdem registriert der Service das IoT-Gerät direkt automatisch beim Cloud-Dienst.



NXP regelt die Geräteberechtigungen.

Dieser flexible Dienst unterstützt mehrere Arten von Berechtigungsnachweisen und wendet je nach Projekt unterschiedliche Konfigurationen an. Berechtigungsnachweise (Device Credentials) können erneuert oder zu im Feld freigegebenen Geräten hinzugefügt werden. Mit der Inbetriebnahme von EdgeLock SE050 und EdgeLock 2GO erhalten Anwender eine End-to-End-Lösung, die einfach, sicher und flexibel ist. Mit der zunehmenden Verbreitung des IoT steigen auch die Risiken. Die EdgeLock-Kombination von NXP mit ihrer hardwarebasierten Sicherheit und dem Service für die Verwaltung von Berechtigungsnachweisen bietet Geräteherstellern Sicherheit in ihrem Geschäft. Wenn NXP EdgeLock die Bereitstellung eines Geräts unterstützt, verkürzt sich nicht nur die Zeit bis zur Markteinführung, es werden auch die täglichen Kosten für den Betrieb einer IoT-Bereitstellung gesenkt, während gleichzeitig die Gewissheit besteht, dass die Geräte durch ein hohes Maß an Sicherheit geschützt sind.

Entfesseln Sie die Kraft: Mehr Geschwindigkeit und verbesserte Effizienz

Das System-on-Chip i.MX 8M Mini [5] ist der erste eingebettete Multicore-Anwendungsprozessor von NXP, der in der fortschrittli-

chen 14LPC FinFET-Prozesstechnologie gefertigt wird und höhere Geschwindigkeit und eine verbesserte Energieeffizienz bietet. Die i.MX 8M Mini-Familie von Anwendungsprozessoren vereint High-Performance-Computing, Energieeffizienz und eingebettete Sicherheit für die schnell wachsende Zahl an Edge-Node-Computing-, Streaming-Multimedia- und Machine-Learning-Anwendungen.

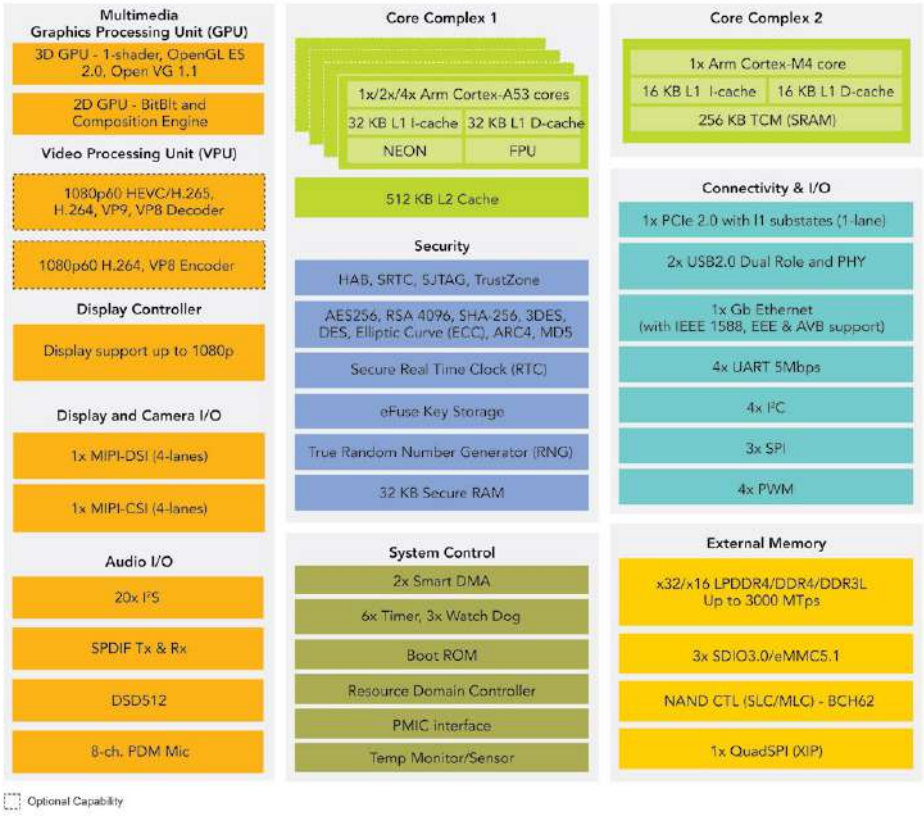
Das System-on-Chip i.MX 8M Mini wird in Single-, Dual- und Quadcore-Varianten mit Arm® Cortex®-A53 angeboten, die mit bis zu 1,8 GHz pro Kern arbeiten. Der in einem fortschrittlichen Low-Power-Prozess gefertigte Kernkomplex ist für lüfterlosen Betrieb, niedrige thermische Systemkosten und lange Batterielebensdauer optimiert. Die Cortex-A-Kerne können abgeschaltet werden, während das Cortex-M4-Subsystem eine stromsparende Systemüberwachung in Echtzeit durchführt. Der DRAM-Controller unterstützt 32-Bit/16-Bit-LPDDR4-, DDR4- und DDR3L-Speicher und bietet damit eine große Flexibilität beim Systemdesign.
Die Kern-Optionen sind beim i.MX 8M Mini für einen extrem niedrigen Stromverbrauch optimiert, der in bestimmten Anwendungen sogar unter einem Watt liegt, und bieten dennoch die nötige Rechenleistung für Consumer-, Audio- und Industrie-Anwendungen sowie beim Machine-Learning-Training und Inferencing bei einer Reihe von Cloud-Anbietern. Das SoC i.MX 8M Mini bietet außerdem Hardware-Beschleunigung für 1080p-Videos, um Zwei-Wege-Videoanwendungen zu ermöglichen, 2D- und 3D-Grafiken für ein reichhaltiges visuelles HMI-Erlebnis sowie fortschrittliche Audio-Funktionen. Eine umfangreiche Auswahl an Hochgeschwindigkeitsschnittstellen ermöglicht eine breitere Systemkonnektivität und zielt auf eine Qualifizierung auf industrieller Ebene ab.
Die Anwendungsbeispiele umfassen:

Industrielle Automatisierung
Der Portenta X8 kann dann als Multiprotokoll-Gateway fungieren und Daten über WLAN, LoRa, NB/IoT, LTE Cat.M1 an die Cloud oder das Enterprise-Resource-Planning-System senden. Durch die Verfügbarkeit von Linux-Containern wie ROS innerhalb der Arduino-Umgebung eignet sich der Portenta X8 hervorragend für autonome, fahrerlose Fahrzeuge.

Gebäude-Automatisierung
Im Zusammenspiel mit intelligenten Umgebungssensoren ermöglicht Portenta X8 die Implementierung von Echtzeit-ML und Bildverarbeitung „on the edge". Intelligente Kioske nutzen in der Regel mehrere Komponenten wie Kartenleser, Kameras oder Mikrofone, was eine vielfältige Auswahl an I/Os erfordert. In Kombination mit einem Max-Carrier gewährleistet der Portenta X8 WLAN-Konnektivität und ermöglicht es Administratoren, die Nutzung der Geräte aus der Ferne zu überwachen. Der Portenta X8 kann gleichzeitig Klimasysteme steuern, intelligente Geräte ein- und ausschalten, die Beleuchtung autonom regeln und die Zugänge kontrollieren.

Beginnen Sie noch heute Ihre Entwicklung mit dem industrietauglichen, sicheren Portenta-X8-SoM [6] mit herausragender Rechenleistung.

(20576-02)RG



Blockschaltung des Anwendungsprozessors i.MX 8M Mini.

WEBLINKS

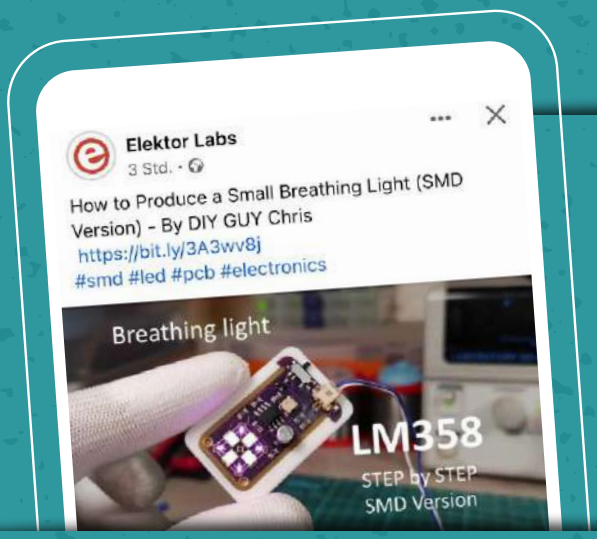
[1] Foundries.io: https://foundries.io/
[2] Arm SystemReady: www.arm.com/architecture/system-architectures/systemready-certification-program
[3] EdgeLock SE050: https://bit.ly/EdgeLockSE050
[4] EdgeLock 2GO: https://bit.ly/EdgeLock2GO
[5] i.MX 8M Mini: https://bit.ly/iMX8MMini
[6] SoM Portenta X8: www.arduino.cc/pro/hardware/product/portenta

Neues *Arduino*- oder *Elektronikprojekt*?

Teilen Sie es mit unserer Community

Folgen Sie uns auf:

www.twitter.com/ElektorDE



www.instagram.com/elektorlabs

www.facebook.com/ElektorDE

elektor

Q&A mit Daria Baradel

Das menschliche Element in der Produktion

▲ *Daria Baradel (vorne) und ihre Arduino-Kollegen.*

Treffen Sie Daria Baradel, die Verantwortliche für die Produktion bei Arduino

Von Keith Jackson (Arduino)

Wie hat einer der beliebtesten Elektronikhersteller der Welt die Herausforderungen der Lieferketten in den letzten drei Jahren erfolgreich gemeistert? Mit einer unerschütterlichen Leidenschaft für Innovation, einer engen Zusammenarbeit mit lokalen Partnern und einer menschlichen Note!

Keith Jackson: Hallo, Daria! Bei allem, was wir in den Nachrichten über die weltweite Verknappung von elektronischen Bauteilen hören, die die Produktion in Fabriken auf der ganzen Welt zum Erliegen bringt, muss es ein herausforderndes Jahr für die Produktion von Arduino gewesen sein.
Daria Baradel: Ja. Jeder hat schon von den Engpässen in der Elektronikindustrie gehört, die alle Produkte betreffen, von neuen Autos bis hin zu Spielkonsolen. Es ist aber nicht nur der Mangel an Lagerbeständen, der die größten Schwierigkeiten verursacht, es sind die unvorhersehbaren Verzögerungen und die ständig steigenden Kosten, die es praktisch unmöglich machen, die Kontrolle zu behalten.

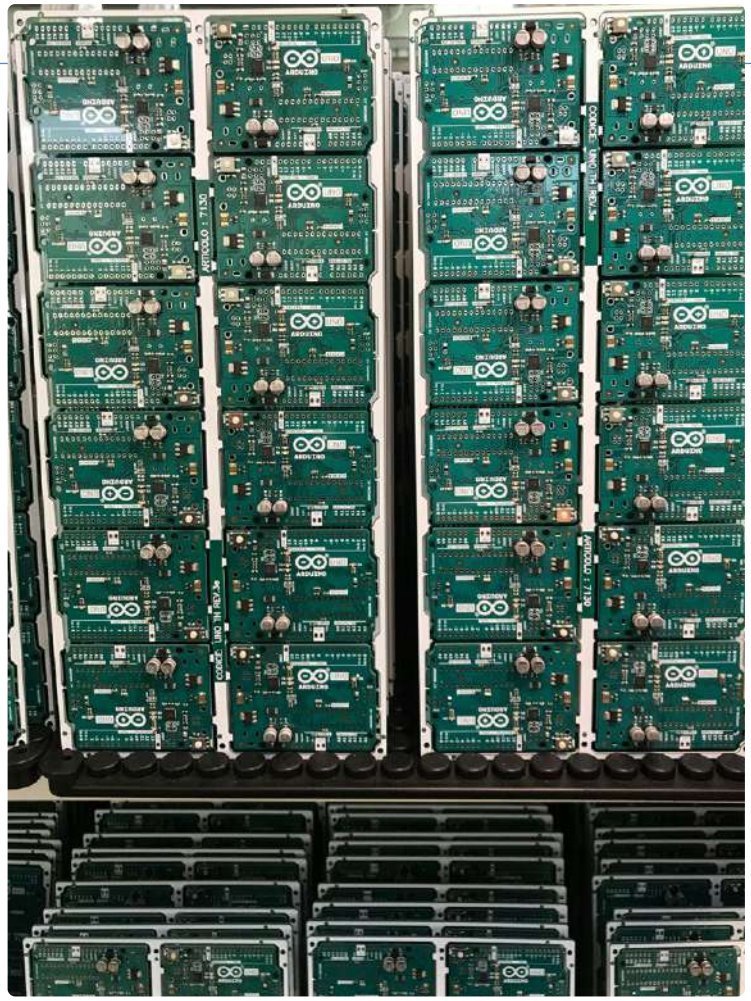
Das gesamte Team musste unseren Ansatz anpassen und flexibler werden. Um die Kontrolle zu behalten, sind Vorhersehbarkeit und Risikomanagement erforderlich, aber wenn die Vorhersehbarkeit nicht mehr gegeben ist, wird die Aufrechterhaltung der Produktion zu einer täglichen Herausforderung. An jedem beliebigen Tag kann eine erwartete Lieferung storniert werden, was zu Produktionsstopps in der Fabrik führt. Aber nicht immer sind es Stornierungen, die zu kurzfristigen Änderungen in der Produktionsplanung führen. Es gibt auch das genaue Gegenteil, wenn plötzlich eine zufällige Lieferung von Bauteilen eintrifft, die erst für 2023 erwartet wurde. In Fällen, in denen dies das einzige noch fehlende Produkt in einer Stückliste war, bietet sich uns plötzlich die Möglichkeit, ein Modell wieder in Produktion zu bringen. Wir passen die Pläne also ständig an das an, was verfügbar ist und was unsere Kunden verlangen. Wir mussten alle Standardverfahren überprüfen und sie in dieser instabilen und unvorhersehbaren Situation anpassen, indem wir sie weniger standardisiert und flexibler gestalteten, um für jedes einzelne Problem, das sich ergeben hätte, gerüstet zu sein.

Keith: Können Sie uns einen Einblick geben, wie Sie und das Team Ihren Ansatz zur Bewältigung dieser Herausforderungen angepasst haben?
Daria: Am Anfang habe ich versucht, vorherzusagen, was passieren würde, aber das hat sich als unmöglich erwiesen. Daher musste ich meine Denkweise ändern und einen proaktiven Ansatz verfolgen. Wir stimmten als Team überein, dass mehr Kontrolle über die Lieferungen notwendig war. Deshalb überprüfen wir jeden Freitag bei allen unseren Lieferanten, was in der nächsten Woche geliefert werden soll. Bei den Hauptkomponenten verfolgen wir dies sogar täglich. Es geht aber nicht nur darum, wie wir mit unseren Zulieferern zusammenarbeiten, sondern es wurde auch dringend notwendig, viel enger mit dem Arduino-Hardware-Team zusammenzuarbeiten, das für die Entwicklung und das Testen der Produkte verantwortlich ist. Bevor eine Stückliste für ein neues Produkt fertiggestellt oder überhaupt bestätigt wird, welche Komponenten während des Prototypings auf einer Platine montiert werden sollen, treffen sich beide Teams nun, um die Verfügbarkeit und Zuverlässigkeit des Lieferanten als Teil des Designprozesses zu bewerten.
Die Verfügbarkeit eines Bauteils und seine Tauglichkeit werden inzwischen beim endgültigen Design berücksichtigt. Es ist aber nicht alles nur negativ, denn das Team hat jetzt ein viel besseres Verständnis für die Platinen und die Bauteile, da es viel früher in den Entwicklungsprozess einbezogen wird. Wir alle erleben also jedes neue Produkt vom ersten Konzept an.

Im Jahr 2022 hat das Lieferketten-Team zusammen mit dem Hardware-Team die Validierung von mehr als 100 alternativen Bauteilen durchgeführt und das Design von 16 Produkten überarbeitet, um die Stückliste anzupassen und mit der Fertigung fortfahren zu können.

Keith: Wie war die Zusammenarbeit mit den Zulieferern, wenn man bedenkt, dass Arduino nur einer von vielen Kunden ist, die nach Bauteilen verlangen?
Daria: Ironischerweise hatte ich in den letzten Jahren viel Zeit in die Automatisierung möglichst vieler Einkaufsverfahren für die Bestellung von Komponenten investiert. In diesem Jahr ist jedoch die menschliche Interaktion wieder in den Vordergrund getreten. Mit anderen Menschen zu kommunizieren und zusammenzuarbeiten ist unverzichtbar. Nur durch ständige Telefonate sind wir in der Lage, mit den neusten Änderungen und den sich daraus ergebenden Konsequenzen Schritt zu halten. Mit den vielen täglichen Nachrichten, die ich verschicke, könnte man mich für einen Stalker halten!
Viele der Arduino-Lieferanten haben ein großes Interesse am gesamten Arduino-Konzept, und nicht wenige von ihnen sind Arduino-Enthusiasten. Daher ist es klar ersichtlich, dass wir von einigen Zulieferern großartige Unterstützung erhalten haben und von anderen weniger. Im Großen und Ganzen sind diejenigen, von denen wir viel Unterstützung erhalten haben, jetzt wirklich treue Arduino-Zulieferer, die eng mit uns als Teil einer großen Gemeinschaft zusammenarbeiten.

Keith: Wie läuft die Produktion und was sind Ihre Aussichten für die Zukunft?
Daria: Trotz allem, was in diesem Jahr passiert ist, liegt das Produktionsvolumen immer noch um mehr als 20 % höher als im gleichen Zeitraum 2021. Dies wurde durch eine vollständig teamübergreifende Leistung erreicht. Das Hardware-Team hat einen enormen Beitrag geleistet, indem es buchstäblich Hunderte von alternativen Komponenten geprüft und genehmigt hat. In vielen Fällen wurden sie noch am selben Tag montiert und zur Genehmigung getestet, um sicherzustellen, dass wir den verfügbaren Bestand beschaffen konnten. Obwohl es noch kein Licht am Ende des Tunnels gibt, was die Bauteileknappheit betrifft, und sich viele Lieferzeiten bis Ende 2023 hinziehen, bin ich also dennoch zuversichtlich, dass wir weiter wachsen können, denn schließlich werden Unternehmen von Menschen gemacht. Die Veränderungen in der Denkweise und den Verfahren, die wir in diesem Jahr einführen mussten, werden uns in Zukunft nur noch stärker und effektiver machen.



Keith: Sie erwähnten, dass Sie alternative Komponenten noch am selben Tag testen und genehmigen. Wie ist das machbar?
Daria: Die Tatsache, dass unsere Hersteller in der Nähe der Arduino-Forschungs- und Entwicklungslabors angesiedelt sind, war während der COVID-Pandemie und in diesen Zeiten der Knappheit ein großer Vorteil. Arduino war schon immer stolz darauf, in Italien zu produzieren. Alle Platinen werden in zwei Fabriken in der Region Piemont in Italien hergestellt. In der Praxis bedeutete dies, dass die Ingenieure bei Problemen sofort in ein Auto sprangen und zur nahe gelegenen Fabrik in Turin fuhren, um die Bauteile zu überprüfen und gegebenenfalls auszutauschen. Wenn wir über die Verfügbarkeit eines alternativen Bauteils informiert werden, müssen wir den Auftrag noch am selben Tag bestätigen, da sonst der Bestand von jemand anderem übernommen wird. Da muss man in Windeseile reagieren. Also fahren die Ingenieure ins Werk, testen das Alternativprodukt und geben uns grünes Licht, dass der Auftrag noch am selben Tag erteilt werden kann. Das wäre natürlich nicht möglich gewesen, wenn unsere Fertigung im Ausland stattgefunden hätte.

Keith: Arduino muss eines der wenigen Elektronikunternehmen sein, das seine Platinen noch von lokalen Herstellern fertigen lässt.
Daria: Ja. Die Herstellung vor Ort war eine bewusste Entscheidung, da Arduino schon immer einen positiven Ansatz in Bezug auf Nachhaltigkeit, Umweltauswirkungen und die lokale Gemeinschaft verfolgt hat. Unsere Nachhaltigkeitsziele sind mehr als nur ein Stück Papier: Wir haben immer versucht, unseren CO_2-Fußabdruck so gering wie möglich zu halten und Abfall zu vermeiden. So verschenken wir beispielsweise Altbestände an örtliche Schulen, und die Herstellung und Verpackung der Produkte erfolgt keine 50 km von unserem Büro in Turin entfernt. Bei der Auswahl unserer Partner und bei der Herstellung der Produkte vor Ort wird der Gemeinschaftsgedanke berücksichtigt. Die Montage und das Verpacken erfolgen durch ein lokales Unternehmen, das aktiv Menschen mit Behinderungen beschäftigt und ihnen die Möglichkeit gibt, unabhängig zu sein.

Keith: Es klingt, als ob Sie sehr stolz darauf sind, bei Arduino zu arbeiten. Aber was inspiriert Daria außer dem täglichen Umgang mit dem Mangel an Bauteilen noch?
Daria: Ich bin gelernte Ingenieurin und habe zwei Masterabschlüsse in Produktionstechnik und Ingenieurmanagement. Als ich im Jahr 2014 als Projektmanagerin ins Unternehmen kam, produzierten wir zwischen 3.000 und 5.000 Platinen pro Monat. Inzwischen sind wir auf das 50-fache angewachsen - das heißt, ich habe jetzt ein Team von acht Mitarbeitern, um das zu managen.
Neben der Arbeit reise ich jedoch gerne und treibe viel Sport, vor allem Klettern, Laufen, Windsurfen und Beachvolleyball. Sportliche Aktivitäten sind für mich der beste Ausgleich zum Alltagsstress. Aber wenn ich einen Traum für die Zukunft hätte, dann wäre es, eines Tages einen Bauernhof zu besitzen, denn ich liebe Tiere, da ich aus einer Familie komme, die Kühe, Schafe und Ziegen gehalten hat. Ich bin sicher, dass ich auch dann überall Arduinos haben werde, um alles so effizient wie möglich zu machen.

(220426-02)SG

Über den Autor

Keith Jackson arbeitet im Marketing von Arduino beschäftigt und begeistert sich für alles, was mit Arduino zu tun hat, denn Arduino ist mehr als ein Unternehmen oder eine Marke, es ist eine ganze, vielfältige Gemeinschaft.

Haben Sie Fragen oder Kommentare?

Haben Sie irgendwelche Fragen oder Kommentare zu diesem Artikel? Wenden Sie sich bitte an den Autor unter k.jackson@arduino.cc oder an die Elektor-Redaktion unter redaktion@elektor.de.

Elektor
Webinars

FEB
2023
16:00 CET

MicroPython Enters the World of Arduino

with Stuart Cording & Sebastian Romero

MicroPython has made it to the world of Arduino, providing the first significant alternative to programming in C and C++. So, what's all the fuss, how easy is it to use, and who can benefit from programming in this, for microcontrollers, relatively new language? Stuart Cording will speak with Sebastian Romero (Head of Content, Arduino) during our live webinar to find out more.

Join for free

www.elektor.com/webinar-MicroPython

elektor

(Source: Shutterstock)

Development Boards

Vergangenheit, Gegenwart und Zukunft

Von Mark Patrick (Mouser Electronics)

In den letzten Jahren ist die Bedeutung des Begriffs „Development Board“ in den unzähligen Bezeichnungen untergegangen, die für Hardware-Boards zu Entwicklungszwecken verwendet werden, wie beispielsweise „Demo-Boards“, „Evaluierungskits“ und „Referenzdesigns“. In diesem Artikel erklären wir die Bedeutung des Begriffs „Development Board“ und zeigen, wie sie sich von ihrem nahen Verwandten, dem Single Board Computer (SBC), unterscheiden. Wir beschreiben ihre Entwicklung von der Vergangenheit bis zur Gegenwart und untersuchen einige Trends, wie sie sich in Zukunft weiterentwickeln könnten.

Was ist ein Development Board?

Zu Beginn müssen wir klar definieren, was unter einem Development Board zu verstehen ist und wie es sich von einem Single Board Computer (SBC) unterscheidet. Ein Development Board wird in der Regel vom Hersteller eines Mikrocontrollers entwickelt, um dessen Funktionen zu präsentieren (obwohl der Begriff inzwischen auch häufig für andere Arten von Komponenten verwendet wird). Ein Mikrocontroller ist ein integrierter Schaltkreis, der einen Prozessor, etwas RAM und einen Flash-Speicher enthält und über E/A-Funktionen verfügt, mit denen er mit der realen Welt interagieren kann. Er arbeitet im Grunde wie ein Mini-Computer in einem einzigen Gehäuse und soll Entwicklern eine bequeme Möglichkeit zur Steuerung von externen Komponenten wie Lichtern, kleinen Motoren und so weiter an die Hand geben. Ein SBC bietet ebenfalls diese Funktionalität, mit dem wesentlichen Unterschied, dass die CPU, der Arbeitsspeicher und der Speicher jeweils in separaten ICs auf der Platine untergebracht sind und über Schnittstellen mit einer Tastatur und/oder einem Display verbunden werden können.

Der Mikroprozessor auf einem SBC benötigt ein Betriebssystem, während ein Mikrocontroller mit einer vom Hersteller bereitgestellten integrierten Entwicklungsumgebung (IDE) verwaltet wird. In vielen Fällen stellen die Hersteller inzwischen Development Boards her, die zwar einen Mikrocontroller enthalten, deren Hauptzweck aber nicht darin besteht, die Funktionen des Mikrocontrollers selbst zu demonstrieren, sondern die der Sensoren oder anderer integrierter Schaltungen, mit denen er verbunden ist. Diese Boards werden als „Demo-Boards“, „Evaluierungskits“ oder – wenn sie so zusammengesetzt sind, dass die Teile einen konkreten Zweck erfüllen – als „Referenzdesigns“ bezeichnet.

Manche Boards dienen nicht in erster Linie der Hardwareentwicklung, sondern bieten Zugang zu den realen Daten, mit denen Softwareentwickler Algorithmen für Anwendungen in den Bereichen Künstliche Intelligenz und Machine Learning erstellen und weiterentwickeln können. Diese Boards entsprechen zwar nicht der ursprünglichen Definition und dem Zweck eines „Development Boards“, werden aber heute allgemein als jedes Stück Hardware verstanden, das für die Hard- und Softwareentwicklung neuer elektronischer Produkte verwendet werden kann.

Vergangenheit

Im Jahr 2006 kam das erste Development Board für Mikrocontroller auf den Markt, das die Aufmerksamkeit der Entwicklergemeinde auf sich zog. Diese Prototyping-Plattform, die später unter dem Namen Arduino [1] (**Bild 1**) bekannt wurde, wurde schnell von vielen Elektronikentwicklern, darunter Enthusiasten, Bastler und Heimwerker, angenommen. Arduino legte den Grundstein für den kommerziellen Erfolg späterer SBCs und mikrocontrollerbasierter Plattformen und wurde im Jahr 2008 vom sogenannten Beagleboard [2] abgelöst, das den Entwicklern eine preisgünstige, von der Community unterstützte Open-Source-Entwicklungsplattform bot. Im Jahr 2012 kam mit dem Raspberry PI [3] der erste Einplatinencomputer auf den Markt. Wie das Beagleboard war auch er als Lernplattform konzipiert, mit der Schüler und Studenten kostengünstig lernen konnten, wie man Programmcode schreibt. Doch der Raspberry PI begeisterte nicht nur Studenten, sondern wurde schnell von Hobbybastlern wie auch von professionellen Entwicklern angenommen.

Gegenwart

Heute gibt es zwei Hauptkategorien von SBCs: Proprietäre und Open-Source-SBCs. Proprietäre SBCs sind in der Regel für den Einsatz in Endanwendungen konzipiert und wurden den gleichen Tests und der gleichen Qualitätssicherung unterzogen wie andere Endprodukte. Sie werden entweder in elektronische Geräte integriert oder in einer Rackmount-Konfiguration installiert. Bei Open-Source-SBCs haben die Benutzer Zugriff auf das Hardware-Design und -Layout sowie auf den Quellcode, den sie verwenden. So können sie schnell und einfach lernen, wie Software und Hardware funktionieren, und das Design dann an ihre Anforderungen anpassen.

Development Boards und SBCs werden heute mit vielen verschiedenen Prozessortypen angeboten. Diese reichen von X86-basierten Prozessoren im traditionellen PC-Bereich (AMD und Intel) bis zu ARM-Prozessoren, die in industriellen und mobilen Anwendungen eingesetzt werden. Dabei kommen Linux und seine Derivate (Ubuntu, Fedora, Debian und so weiter), Android und Windows CE als Betriebssysteme auf SBCs am häufigsten zum Einsatz. Mikrocontroller-Development-Boards benötigen kein Betriebssystem und werden über eine vom Hersteller bereitgestellte IDE programmiert. Mittlerweile verfügen sowohl Mikrocontroller-Development-Boards als auch SBCs über drahtlose Konnektivität (WLAN, Bluetooth) und die neuesten Audio- und Videoschnittstellen. Somit bieten einige SBCs inzwischen Funktionen, die mit denen vieler PCs und Tablets vergleichbar sind.

Zukunft: Development Boards werden zum Endprodukt

In der Vergangenheit haben die Hersteller ihre Entwicklungstools mit der Absicht entwickelt, sie als Marketinghilfe zu nutzen, um die Wahrscheinlichkeit zu erhöhen, ihre Mikrocontroller an potenzielle Kunden zu verkaufen. Das wird in der Branche oft als „Design-in" bezeichnet. Sie versprachen sich davon, dass durch die Minimierung des Arbeitsaufwands für die Entwickler, um ein Produkt im Labor zum Laufen zu bringen, und durch die Erleichterung des Zugriffs und der Analyse seiner Funktionen die Wahrscheinlichkeit steigen würde, dass sie sich bei der Entwicklung von Produktprototypen für ihren Mikrocontroller und die dazugehörigen Bauteile entscheiden. Dies sollte schließlich zu höheren Bestellmengen führen, wenn das entsprechende Bauelement für die Massenproduktion ausgewählt wurde. Dieser Ansatz ist bei Produkten sinnvoll, bei denen der Unterschied in den technischen Spezifikationen zwischen den Bauteilen verschiedener Anbieter vernachlässigbar ist. Für die Hersteller ist dieser Ansatz jedoch in gewisser Weise ein Opfer seines eigenen Erfolgs geworden. Sie haben erkannt, dass sie den Arbeitsaufwand für einen Entwickler, der sich mit ihren Produkten beschäftigt, noch weiter reduzieren müssen. Somit ist das Development Board zum wichtigsten Unterscheidungsmerkmal geworden, vor allem bei Produkten, die denen der Konkurrenz sehr ähnlich sind.

Die Erwartungen der Entwickler sind so gestiegen, dass sie selbst bei Bauteilen, die einen klar erkennbaren Wettbewerbsvorteil haben (etwa bei der Leistung oder der Geschwindigkeit), erwarten, dass die zugehörigen Development Boards Plug'n'Play-fähig sind.

Mit Referenzdesigns, die aus einem Mikrocontroller und anderen integrierten Schaltkreisen (in der Regel Sensoren) bestehen, haben die Hersteller ihr Leistungsangebot weiter verbessert. Ursprünglich sollten diese Referenzdesigns nur einen Leitfaden dafür bieten, wie die einzelnen Bauelemente miteinander verbunden werden können und damit die elektrische Funktionalität des Endprodukts nachbilden. Der Formfaktor, die Größe des Designs und die Einfachheit der Fertigung standen dabei im Hintergrund. Einige Hersteller sind jedoch mit ihren Referenzdesigns einen Schritt weiter gegangen

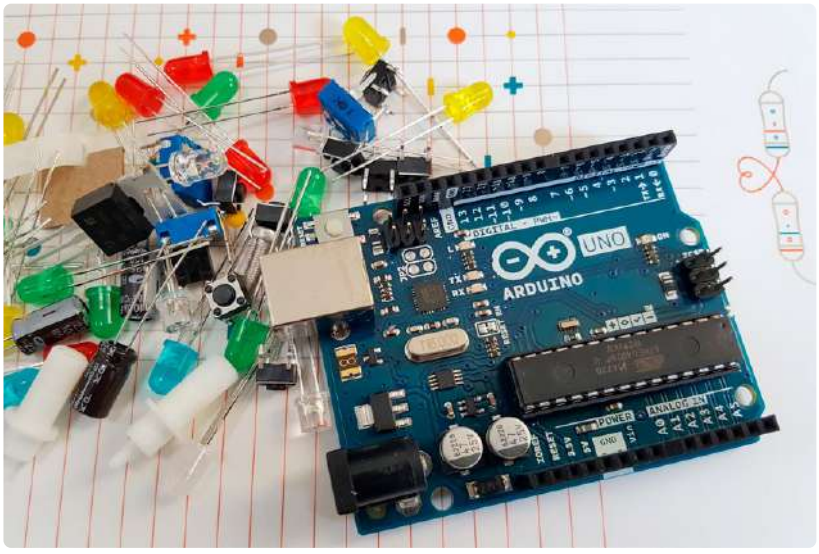


Bild 1. Mikrocontroller-Development-Board „Arduino".(Source: Shutterstock)

▼

▲

Bild 2. HSP3.0 von Maxim Integrated.

und haben vollwertige Produktprototypen und sogar voll funktionsfähige Produkte entwickelt.
Die Referenzdesigns der Health Sensor Platform (HSP) [4] von Maxim Integrated (jetzt Teil von Analog Devices) sind ein gutes Beispiel für diese Entwicklung. Die erste Version dieser Referenzdesigns war ein kleines Development Board mit einer Reihe von Sensoren (Temperatur, Druck, Beschleunigungsmesser, Biopotenzial et cetera), die für den Einsatz in Gesundheits- und Fitnessanwendungen geeignet waren und mit einem Mikrocontroller konfiguriert werden konnten. Die beiden Nachfolger, HSP2.0 und HSP3.0, hatten Formfaktoren. Sie konnten am Handgelenk getragen werden und sehen anderen auf dem Markt erhältlichen Wearables sehr ähnlich (**Bild 2**). Die Entwickler konnten dadurch die Funktionalität ihrer Sensoren in der Praxis testen. Ein wichtiger Vorteil dieser Designs war auch, dass Softwareentwickler freien Zugang zu den Sensormesswerten hatten (Informationen, die bei anderen Gesundheits- und Fitness-Wearables nicht so leicht zugänglich sind). Mit diesem Ansatz konnten Algorithmen für Maschinelles Lernen und Künstliche Intelligenz entwickelt werden, die einen Mehrwert für die Anwendung darstellen.

Bild 3. Speicherprogrammierbare Steuerung. (Source: Shutterstock)

▼

Maxim wollte demonstrieren, dass seine Hardware einen einfachen Zugang zu den Daten ermöglicht, und hoffte, dass Produktentwickler einige (oder alle) der ICs in seiner Sensorlösung für den Einsatz in ihren Produkten auswählen würden. Dieser Ansatz ging so weit, dass Maxim das MAX HEALTH BAND [5] (Armband) und den MAX ECG MONITOR [6] (Brustgurt) entwickelte, die beide als voll funktionsfähige Gesundheits- und Fitness-Wearables konzipiert und produziert wurden. Sie waren zwar nicht für den direkten Verkauf an Verbraucher bestimmt, aber Unternehmen konnten gegen eine Lizenzgebühr einen Vertrag mit Maxim abschließen, um diese Produkte unter ihrem eigenen Label zu vermarkten.
Indem ein voll funktionsfähiges Produkt auf diesem Wege angeboten wird, bei dem die gesamte Entwicklungsarbeit bereits geleistet wurde, kann eine neue und breitere nicht-technische Kundenbasis angesprochen werden. Das Thingy:91 [7] von Nordic Semiconductor ist ein weiteres Beispiel für eine Entwicklungsplattform, bei der die Hardware fast zur Nebensache geworden ist. Die Entwickler müssen die Software und die Algorithmen entwickeln, mit denen die eigentliche Leistungsfähigkeit der Hardware optimal ausgeschöpft werden kann (und die dadurch gleichzeitig zur offensichtlichen Wahl für die Verwendung in neuen Produktdesigns wird, die diese Algorithmen nutzen). Es ist davon auszugehen, dass dieser Ansatz in Zukunft von noch mehr Herstellern übernommen werden wird.

Vermehrter Einsatz von Development Boards in Industrieprodukten

Development Boards und SBCs werden immer häufiger für den Einsatz in kommerziellen Produkten angepasst. Doch ein weiterer Trend ist ihr Einsatz in Anwendungen mit geringerem, aber gleichzeitig höherwertigem Volumen: industrielle Endprodukte wie Speicherprogrammierbare Steuerungen, die strengeren Standards unterliegen als ihre kommerziellen Pendants (**Bild 3**).

Testboards für industrielle Anwendungen

Bei vielen der heutigen SBCs handelt es sich um vollständig verifizierte Designs, da die darin enthaltenen Teile ursprünglich für den Einsatz in Endprodukten entwickelt wurden und daher getestet und qualitätsgesichert sind. Hinzu kommt, dass Open-Source-Designs ständig von zahlreichen kompetenten Designern und Programmierern überprüft werden, die die Boards und die verwendete Software aktualisieren und bewerten.
Die Tests von SBC-Platinen werden jetzt von renommierten Design- und Fertigungsunternehmen durchgeführt. Dabei unterliegen sie den gleichen strengen Qualitätskontrollen wie alle anderen Endprodukte, sodass sie sogar eine CE- oder FCC-Zertifizierung erhalten können. Dieser Testablauf lässt sich leicht auf die Anforderungen von Industrieprodukten ausweiten.
Demgegenüber werden die von Herstellern oder Drittanbietern gelieferten Development Boards für

Mikrocontroller, die zwar in der Regel für den Einsatz in kommerziellen Produkten geeignet sind, meist nicht denselben strengen Tests unterzogen, die für industrielle Produkte vorgeschrieben sind. Das bedeutet, dass die Hersteller sie gegenwärtig nicht für den sofortigen Einsatz (in ihrer jetzigen Form) in diesen Anwendungen empfehlen.

Auch wenn einige Boards mit Komponenten in Industriequalität bestückt sind, handelt es sich häufiger um kommerzielle Boards, die nur für den Betrieb bei Raumtemperatur ausgelegt sind. Prototypen von Development Boards werden in der Regel mehrere Tage oder Wochen lang bei Raumtemperatur getestet, wobei dies je nach Hersteller variiert, da es keine festen Standards gibt. Die primäre Qualitätsanforderung der Hersteller besteht darin, dass ihre Boards bei Raumtemperatur zuverlässig funktionieren. Daher sollten sich Käufer darüber im Klaren sein, dass es unwahrscheinlich ist, dass sie bei extremen Temperaturen oder Feuchtigkeit getestet wurden. Ebenso wenig werden sie üblicherweise daraufhin getestet, ob sie den Belastungen durch starke Vibrationen oder Stöße standhalten.

Daher besteht das Hauptziel bei der Auswahl eines Development Boards für den Einsatz in einer industriellen Anwendung darin, das Risiko zu reduzieren. Bei der Entscheidung für ein Board zur Verwendung in einer industriellen Anwendung müssen alle Komponenten des Boards die richtige Temperaturklasse besitzen. Zudem ist es sinnvoll, mehrere Boards gleichzeitig über mehrere Tage hinweg einem Stresstest bei hohen Temperaturen zu unterziehen. Wenn ein Development Board in einem Produkt eingesetzt werden soll, das hoher Luftfeuchtigkeit ausgesetzt ist, muss es unter vergleichbaren Bedingungen getestet werden. Gleichermaßen sollte ein Board, das für den Einsatz in einer Anwendung mit vielen Vibrationen vorgesehen ist, in einem Testrahmen montiert und auf Vibrationen getestet werden.

Fazit

Mit SBCs und Mikrocontroller-Development-Boards können kleine Unternehmen ihre Designs schnell auf den Markt bringen, ohne die Kosten für die Entwicklung neuer Hardware tragen zu müssen. Dadurch können sie sich auf Software-Innovationen und zunehmend auch auf die Entwicklung von Algorithmen für Machine Learning und Künstliche Intelligenz konzentrieren. Die Einsatzmöglichkeiten von SBCs und Development Boards gehen inzwischen weit über das hinaus, wofür sie ursprünglich gedacht waren, und sie haben die jüngste Geschichte der Elektronikindustrie nachhaltig beeinflusst. SBCs und Development Boards werden immer leistungsfähiger, intelligenter und reaktionsschneller und sind dabei sowohl für professionelle Entwickler als auch für Elektronikliebhaber gleichermaßen leicht zugänglich.

220597-02

Über den Autor

Als Technical Marketing Manager für EMEA bei Mouser Electronics ist Mark Patrick für die Erstellung und Verbreitung technischer Inhalte in der Region verantwortlich - Inhalte, die für die Strategie von Mouser zur Unterstützung, Information und Inspiration des technischen Publikums entscheidend sind.

Bevor er das technische Marketingteam leitete, war Patrick Teil des EMEA-Supplier-Marketingteams und spielte eine wichtige Rolle beim Aufbau und der Entwicklung von Beziehungen zu wichtigen Fertigungspartnern. Neben einer Vielzahl von technischen und Marketing-Positionen war Patrick zuvor acht Jahre bei Texas Instruments im Anwendungssupport und im technischen Vertrieb tätig.

Er ist ein „praktischer" Ingenieur mit einer Leidenschaft für alte Synthesizer und Motorräder und schreckt auch nicht davor zurück, diese zu reparieren. Patrick hat einen erstklassigen Abschluss (Honours Degree) in Elektronikingenieurwesen von der Universität Coventry.

WEBLINKS

[1] Arduino Boards Distributor: https://elektor.link/MouserArduino
[2] BeagleBoard Distributor: https://elektor.link/MouserBeagleBoard
[3] Raspberry Pi Distributor: https://elektor.link/MouserRaspberryPi
[4] Maxim Integrated Distributor: https://elektor.link/MouserMaxim
[5] MAX HEALTH BAND: https://elektor.link/MouserMaxHealthBand
[6] MAX-ECG-MONITOR : https://elektor.link/MouserMaxECGMonitor
[7] Nordic Semiconductor Thingy:91™ Multisensor Prototyping Kit : https://elektor.link/MouserThingy91

Blumenkunst aus Muskeldrähten

Kinetische Skulpturen, die mit Klang kommunizieren

▲ *Bild 1. Die Ergebnisse eines Workshops zum Thema „Kinetische Flora“.*

Von Dave Vondle, IDEO

Seit der Entdeckung ihrer seltsamen Eigenschaften vor über 60 Jahren sind Nitinoldrähte eine Lösung, die nach Anwendungen sucht. Hier ist eine Idee aus dem Bereich der Kunst - Blumen, die mit Licht, Musik und Bewegung auf Klang reagieren.

Dieses Jahr wurden einige von uns bei der Designfirma IDEO eingeladen, einen Workshop auf dem Eyeo-Festival [1] durchzuführen. Wir wollten etwas Physisches mit Menschen bauen, aber etwas, das mit Code gesteuert werden und mit dem man experimentieren kann. Meine Kollegin und Mitarbeiterin Jenna Fizel hat viele anregende Dinge getan, die Software geschrieben, die die Herstellung von 3D-Objekten aus Papier und anderen dünnen Materialien unterstützt. In letzter Zeit habe ich mit Nitinol-Draht experimentiert, einem Material, das seine Form verändern kann, wenn man Strom hindurchleitet. Mit der zusätzlichen Unterstützung der IDEO-Softwaredesigner Derek Olson und YC Sun wollten wir herausfinden, wie wir ein Kit aus Teilen zusammenstellen können, mit dem sich bewegliche Papierskulpturen herstellen lassen (**Bild 1**). Wir wollten, dass die Teilnehmer den Workshop mit etwas ganz Eigenem verlassen, aber auch, dass das Endprodukt des Workshops ein Gemeinschaftsprodukt ist. Dies war der erste persönliche Workshop, den wir nach der Pandemie durchgeführt haben, und wir wollten, dass es für uns alle einen einzigartigen Wert hat, etwas gemeinsam und persönlich zu bauen. Können die von uns geschaffenen Skulpturen miteinander kommunizieren? Wie würden sie kommunizieren? Wir dachten uns, wenn sie klanglich kommunizieren könnten, könnten wir sie auch zusammen spielen hören. Der Workshop sollte eine generative Sinfonie hervorbringen!

Bild 2. Fädeln des Nitinoldrahtes durch die Blumen.

Die Teilnehmer des Workshops verfügten über ein breites Spektrum an Hintergrundwissen und Elektronikkenntnissen. Mit dem Bau dieser Blumen konnten wir die Leute mit neuen, vielschichtigen Konzepten vertraut machen. Einige der Grundlagen waren „Erste Schritte mit Arduino“ und das Ohmsche Gesetz, um die Spannung zu verstehen, die für den Antrieb des Nitinols erforderlich ist. Zu den komplexeren Konzepten, die wir abdecken konnten, gehörten die Pulsbreitenmodulation, die schnelle Fourier-Transformation (FFT) zur Identifizierung der Frequenzen benachbarter Blumen und die Grundlagen der Audiosynthese.

Jeder Teilnehmer malte eine Blume mit selbst gestalteten Blütenblätter-Formen (**Bild 2**). Die Elektronik in den Blumen steuert die LED-Leuchten an der Spitze, die Töne, die die Blume erzeugt, sowie die unabhängige Steuerung der Position der Blütenblätter. Ein Video der endgültigen Aufführung bei einem späteren IDEO-Workshop finden Sie unter [2].

Im Folgenden gehe ich zunächst auf das Design der Hardware und dann auf die Software ein. Zum Schluss möchte ich Ihnen zeigen, wie Sie Ihr eigenes Gerät bauen können.

Die Hardware

Die Hardware basiert aus dem Mikrocontroller-Board Trinket M0 von Adafruit [3] und dem Elektret-Mikrofon-Breakout von Sparkfun [4]. Zusätzlich zu den Funktionen dieser Boards haben wir einen Lautsprecher-Verstärker, eine 5-V-Buck-Boost-Schaltung für den Betrieb an einer Batterie, einige adressierbare WS2812-LEDs und acht MOSFETs zur Steuerung des Stroms durch die Nitinoldrähte hinzugefügt. Der vollständige Schaltplan ist in **Bild 3** dargestellt, die fertige Platine sieht aus wie in **Bild 4a** und **Bild 4b**.

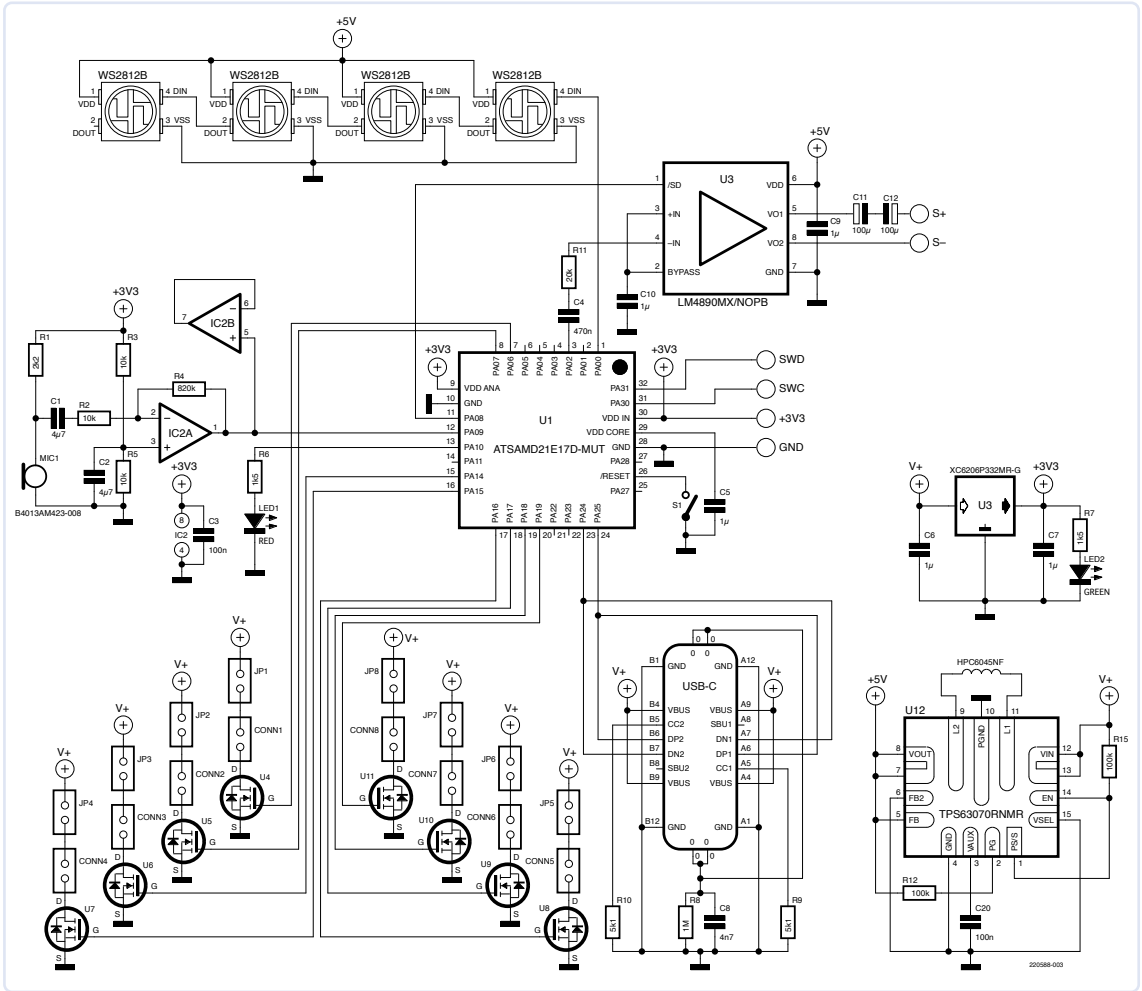


Bild 3. Schaltplan für die Eyeo-Flower.

Bild 4a. Fertige Platte, Bestückungsseite.

Bild 4b. Fertige Platine, Blütenblätter-Seite.

▶

Für unseren Workshop wollten wir Löten vermeiden, da Nitinol ohne spezielle Materialien und Techniken extrem schwierig zu löten ist. Stattdessen verwenden wir ein System aus Gewindebuchsen und Schrauben, um den Draht mechanisch zu verbinden. Auf diese Weise lässt sich auch die mechanische Vorspannung der Drähte in den Blütenblättern genau steuern.
Die in Buchsen gesteckten Widerstände, die den maximalen Strom begrenzen, werden gemäß der Länge des Nitinoldrahtes ausgewählt. Wir verwenden 100-µm-Nitinoldraht [5] mit einem Widerstand von 126 Ω/m und einem maximalen Nennstrom von 200 mA. Mit diesen Angaben kann man einen Widerstand berechnen, der eine Beschädigung des Drahtes verhindert. In ersten Tests hatten wir die Widerstände weggelassen und uns dafür entschieden, den Strom durch die Einstellung eines maximalen PWM-Verhältnisses in der Software zu begrenzen. Wir haben jedoch gelernt, dass während des Programmierens manchmal ein Pin auf „high" ging und den Draht abgefackelt hat, so dass wir dieses Problem mit handfesten Festwiderständen gelöst haben.

Der Mikrocontroller

Wir haben uns für die ATSAMD21-Mikrocontroller-Familie entschieden, weil der Controller bereits in einem einzigen Chip Arduino-Kompatibiltät bietet. Die meisten anderen Arduino-Boards benötigen einen weiteren Baustein für die serielle UART-zu-USB-Umsetzung. Die Variante ATSAMD21E18 im Adafruit Trinket M0 war jedoch aufgrund der Chip-Knappheit nicht verfügbar; wir konnten jedoch verfügbare ATSAMD21E17D-Chips finden. Der Hauptunterschied zum 18er-Chip ist der kleinere Flash-Speicher (128 KB statt 256 KB). Obwohl dieser Unterschied für uns unwesentlich ist, mussten wir den Arduino-Bootloader [6] (von uf2-samdx1 abgekupfert) modifizieren und ein neues Board in der Arduino-IDE erstellen, um mit diesem neuen Chip/Bootloader zu arbeiten.

Ein Codebeispiel

Der Beispiel-Sketch [7] bietet eine Grundlage für das gemeinsame „Singen" der Blumen. Er sorgt für hörbare Töne im Lautsprecher, hört die Noten mit dem Mikrofon ab und reagiert mit Bewegungen der Blütenblätter und Lichtsignalen von den LEDs. Ich werde ein wenig aufschlüsseln, wie das alles funktioniert.

Klänge abspielen

Der ATSAMD21 hat einen Pin, der als DAC (Digital-Analog-Wandler) konfiguriert werden kann. Diese Funktion ist prädestiniert, um eine große Vielfalt an musikalischen Klangfarben aus dem Chip herauszuholen. Damit kann die Blume viel ausdrucksstärker sein, als wenn wir die Arduino-Tone-Library [8] verwenden würden, deren `tone()`-Funktion es nur erlaubt, die Frequenz einer Rechteckwelle zu variieren. Der von uns verwendete DAC erlaubt es aber, die Form der Welle selbst zu steuern. Wir verwenden dazu die fantastische Mozzi-Bibliothek [9], die einen Rahmen für die Definition und Wiedergabe von Klängen aus diesem DAC bietet.
Im Code richten wir ein paar Kosinus-Oszillatoren ein - einen für die Hauptträgerfrequenz und einen für das Vibrato. Außerdem sorgen wir für eine Hüllkurve (auch bekannt als ADSR - Attack, Decay, Sustain, Release), mit der wir die Amplitude des Klangs über die Zeit steuern können.
Während wir die Audiowellenform abspielen, sucht der Mikrocontroller in einer Kalkulationstabelle den nächsten Spannungspegel, der vom DAC gesendet werden muss. Dies geschieht mit einer Frequenz von 16,384 kHz. Wenn aber in dieser Zeitspanne ein blockierender Code oder viele Verarbeitungsvorgänge stattfinden würden, könnte die Erzeugung einer gleichmäßigen Welle gestört werden. Aus diesem Grund hört der Code zuerst zu, reagiert mit Licht und Bewegung und spielt erst dann, wenn keine anderen Prozesse stattfinden, den Ton selbst ab.

Auf den Klang hören

Wie in der Einleitung beschrieben, hört die Blume auf einen bestimmten Ton und reagiert dann mit ihrem eigenen Ton. Das Mikrofonsignal wird über einen Verstärker an einen Pin des Mikrocontrollers geführt, der als „Analog In" konfiguriert werden kann.
Damit die Blume den Ton oder die Frequenz der Töne, die sie hört, verstehen kann, verwenden wir das bereits erwähnte Prinzip der schnellen Fourier-Transformation. Dabei handelt es sich um eine mathematische Transformation, die eine Reihe von Abtastwerten im Zeitbereich (Zeit vs. Amplitude) in eine Reihe von Abtastwerten im Frequenzbereich (Frequenz vs. Amplitude) umwandelt. Dieser mathematische Prozess kann recht ressourcenintensiv sein und geht zu Lasten der Frequenzgranularität und der Geschwindigkeit/Speicherkapazität.

Damit dies so schnell wie möglich abläuft, verwenden wir die beiden Adafruit-Bibliotheken Zero-DMA (Direct Memory Addressing) [10], um Samples vom Mikrofon in ein Array zu ziehen, und Zero-FFT [11], um die FFT durchzuführen. Sobald die Daten durch die FFT verarbeitet wurden, können wir uns die Primärfrequenz des Samples ansehen, um zu überprüfen, ob sie mit der gesuchten Note übereinstimmt.

Bewegen der Blütenblätter

Ein Prototyp der Papierblüte ist in **Bild 5** zu sehen, eine animierte GIF-Ansicht in Aktion finden Sie unter [12]. Der Draht wird asymmetrisch durch das Blütenblatt gezogen. In **Bild 6** ist zu sehen, dass sich auf der Oberseite des Blütenblattes viel mehr Draht befindet als auf der Unterseite. Wenn sich der Draht zusammenzieht, schrumpft die Oberseite des Blütenblatts, was zu einer Scherbelastung auf der Oberfläche führt, die das Blütenblatt krümmt.

Bevor wir uns mit dem Code der Blütenblattbewegung befassen, sollten wir uns einen Moment Zeit nehmen, um die Funktionsweise von Nitinol zu erläutern. Elektrischer Strom erhitzt den Nitinol-Draht über eine Schwelltemperatur hinaus, bei unserem Draht 70 °C. Bei der Schwelltemperatur erfährt das Material eine Phasenänderung und zieht sich zusammen. Damit der Draht wieder seine ursprüngliche Länge annimmt, muss er erneut physisch gedehnt werden.

Der Phasenwechsel folgt diesem Zyklus (**Bild 7**): Im gedehnten Zustand befindet sich der Draht im verformten *martensitischen* Zustand. Die Kristallstruktur von Martensit wird als „kubisch-raumzentriert" bezeichnet. Bei Erwärmung geht der Nitinol in eine *austenitische* Kristallstruktur über, die „kubisch flächenzentriert" wird und „dicht gepackt" ist, was bedeutet, dass die Struktur mehr Atome auf engstem Raum zulässt. Da sich diese Kristallstruktur bei der Erwärmung des Drahts ändert, ordnen sich die Atome in einer dicht gepackten Anordnung neu an, und dies führt dazu, dass sich der Draht zusammenzieht.

Wie wir bereits erwähnt haben, muss der Draht physisch zurückgestreckt werden. Wir brauchten eine Möglichkeit, dies ohne klobige oder komplizierte Federn zu bewerkstelligen. Anstatt Papier für die Blütenblätter zu verwenden, wählten wir ein mattes Polypropylen-„Aquarellpapier" namens YUPO [13]. Durch die dem Material innewohnende Elastizität als Feder werden die Blütenblätter nach dem Biegen wieder in einen offenen Zustand zurückversetzt.

Jedes Blütenblatt kann separat gesteuert werden und ist über einen MOSFET mit einer PWM-Leitung verbunden. Im Code schließen wir die Blütenblätter, und da sie dann „aus" sind, wenn wir sie öffnen, gibt dies unserem Prozessor die Möglichkeit, Töne abzuspielen (auch wenn die Blütenblätter noch geöffnet sind).

Um die Nitinol-Schutzwiderstände zu ermitteln, müssen wir ein wenig rechnen. Bei unserer Blütenblattform haben wir einen etwa 13 cm langen Draht, der von Pfosten zu Pfosten führt. Unser Draht weist einen Widerstand von 126 Ω/m und einen maximalen Nennstrom von 200 mA auf. Unser Drahtwiderstand beträgt also 126 Ω/m × 0,13 m = 16,38 Ω. Um den Widerstand zu berechnen, den wir bei einer maximalen Spannung von 5 V benötigen, verwenden wir das Ohmsche Gesetz U/I = R, also 5 V / 0,2 A = 25 Ω. Wir haben einen 10 Ω-Widerstand in Reihe mit unserem 16,38-Ω-Draht gewählt (10 Ω plus 16,38 Ω ist mehr als 25 Ω), so dass niemals zu viel Strom durch den Draht fließen kann.

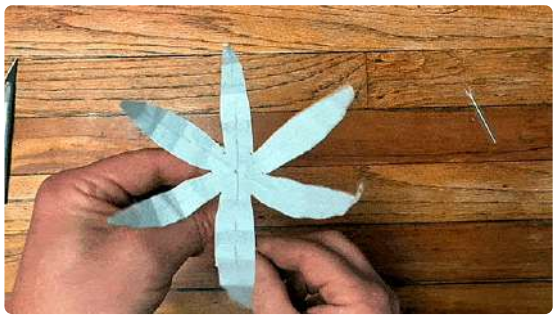


Bild 5. Der erste Prototyp aus Papier mit einem Strohhalm, einer Schnur und Papier für einen Test, bevor die Hardware entworfen wird.

◀

Bild 6. Nahaufnahme eines Blütenblatts mit Draht.

◀

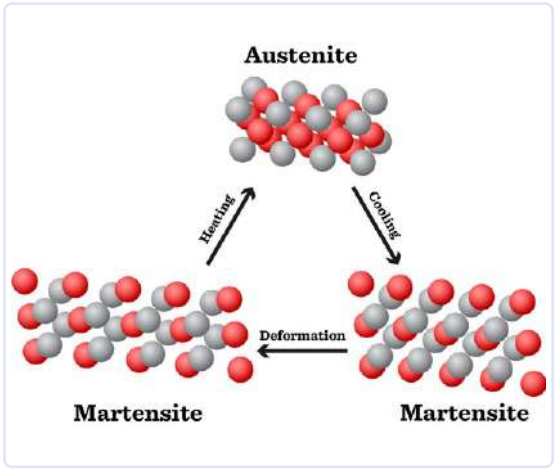


Bild 7. Ein Modell des Nitinol-Kristallstrukturzyklus mit Formgedächtnis.

◀

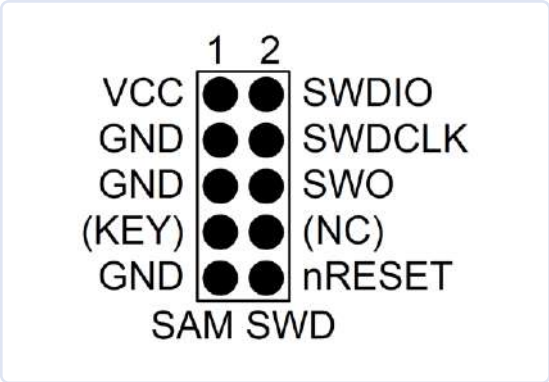


Bild 8. Empfohlene ARM-SWD/JTAG-Anschlussbelegung. Quelle: Atmel-Datenblatt.

LED-Leuchte

Der uf2-Bootloader, den wir modifiziert haben, unterstützt NeoPixel-LEDs des Typs WS2812 der Marke Adafruit. Dadurch können die LEDs auch eine Art Diagnosegerät für den Status des Bootloaders/USB-Anschlusses sein. Um die LEDs zu steuern, wenn der Code läuft, verwenden wir die Adafruit-Bibliothek NeoPixel [14]. WS2812-LEDs sind für Verkettung ausgelegt. Wenn Daten an die LEDs gesendet werden, erhält jede LED 8 Bits pro Farbe (insgesamt also 24 Bits). Wenn wir mehr Daten senden, werden diese Daten auf die nächste LED in der Reihe verschoben, was die unabhängige Steuerung zahlreicher LEDs über einen einzigen Pin ermöglicht.

Eigene Blumen herstellen

Wir haben eine interaktive Website [15] erstellt, auf der Sie schnell eine Blütenblattform skizzieren können, die Sie mit einer Desktop-Schneidemaschine wie einer Cricut ausschneiden können. Eagle-Quelldateien, Gerber-, CPL- und BOM-Dateien zur Herstellung der Leiterplatten finden Sie auf der GitHub-Seite des Projekts [16]. Mit diesen Dateien können Sie über ein Board-Fab-Haus bestückte Platinen erhalten. Wir haben JLCPCB [17] genutzt, daher sind diese Dateien bereits für die PCB- und Bestückungsdienste von JLCPCB formatiert. Sie können natürlich versuchen, sie selbst zu bestücken, aber es gibt eine Reihe von Fine-Pitch-ICs sowie passive Bauteile der Größe 0402, die schon eine gewisse Herausforderung darstellen.

Im GitHub-Repo gibt es auch Links, wo Sie das YUPO-Papier, die Schwanenhals-USB-Kabel, den Nitinol-Draht, die Schrauben und die Gewindeeinsätze finden können. Um den Arduino-Bootloader auf den Chip zu bringen, haben wir ein SEGGER-J-Link-Programmiergerät verwendet und die hervorragende Anleitung zur Programmierung von SAMD-Bootloadern von Adafruit [18] befolgt. Um das Board mit dem Programmiergerät zu verbinden, holen wir Rat bei der SAM-SWD-Pinbelegung aus dem *ICE User Guide* von Atmel [19] (siehe **Bild 8**). Für uns ist wichtig, dass:

Pin 1: VCC / Vref
Pin 2: SWDIO (SWD)
Pin 3: GND
Pin 4: SWDCLK (SWC)

Diese müssen mit den entsprechenden Pins auf der Platine verbunden werden - in **Bild 9** als GND, SWC,

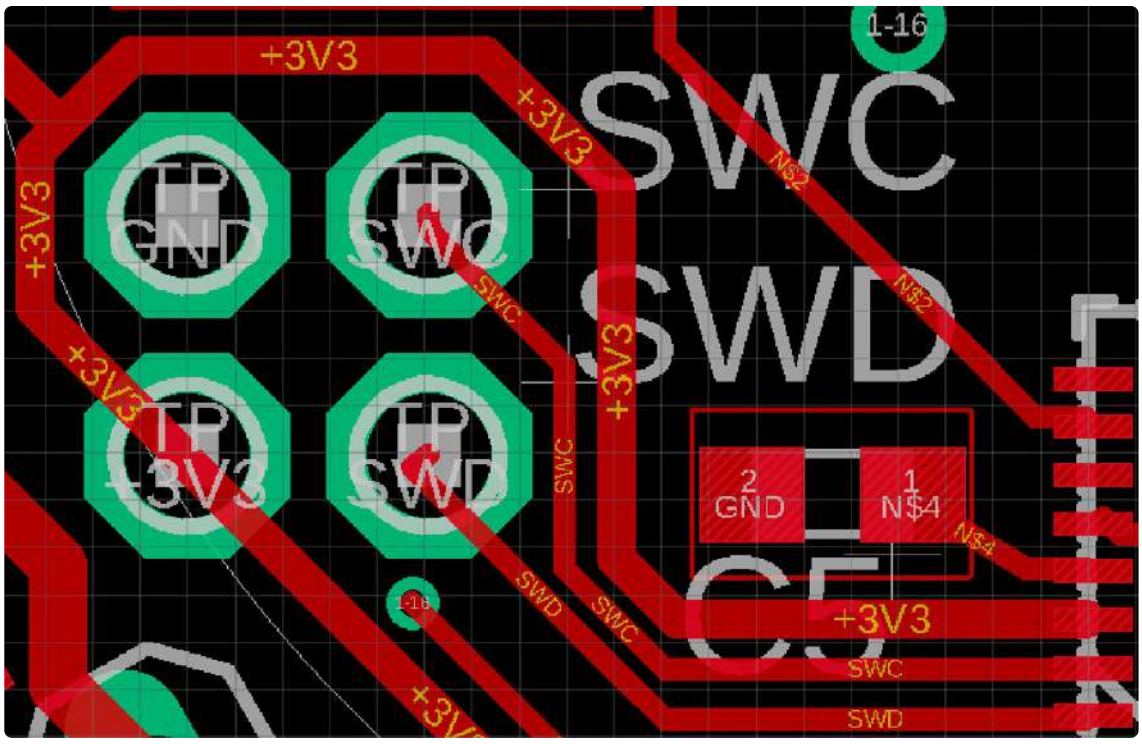


Bild 9. Lage der SWD-Pins auf der Platine.

+3V3 und SWD angegeben. Da wir viele Platinen programmieren wollten, haben wir ein Programmiergerät mit so genannten Pogo-Pins gebaut, um die Platinen leicht austauschen zu können (**Bild 10**).
Wir müssen auch die Gewindeeinsätze in unsere Platine einpressen, um den Nitinol-Draht auf der Platine festschrauben zu können. Wir haben den alten Griff eines Cuttermessers verwendet, um eine maßgeschneiderte Einpressvorrichtung zu bauen (**Bild 11**). Je nach den Ihnen zur Verfügung stehenden Werkzeugen gibt es wahrscheinlich bessere Möglichkeiten zur Herstellung von Tools zum Programmieren und zum Einpressen der Muttern. Lassen Sie es uns wissen, wenn Sie einen besseren Weg finden!

Zum Schluss

Es hat uns Spaß gemacht, diese Tools und Vorrichtungen zu bauen und sie mit der Welt, also Ihnen zu teilen. Wir denken, dass es unterhaltsam ist, mit einer Reihe von verschiedenen Prinzipien zu experimentieren. Wenn Sie darauf aufbauen oder mit uns darüber diskutieren wollen, lassen Sie es uns bitte wissen!

(220588-02)RG

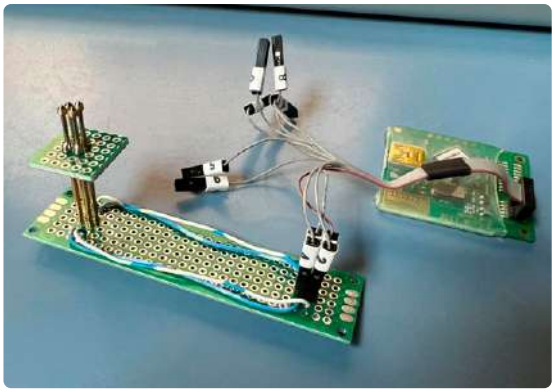


Bild 10. Benutzerdefinierter Programmieradapter.

Bild 11. Selbst entworfenes kleines Presswerkzeuge.

Haben Sie Fragen oder Kommentare?

Wenn Sie auf unserer Entwicklung aufbauen oder mit uns darüber diskutieren möchten, senden Sie eine Nachricht an @ideo auf Instagram, schreiben Sie mir eine E-Mail an dvondle@ideo.com oder kontaktieren Sie die Elektor-Redaktion unter redaktion@elektor.de.

Über den Autor

Dave Vondle ist Director of Experimentation and Publishing bei IDEO. Er schafft gut gemachte Produkte und Erfahrungen, indem er ihre Entwicklung entwirft und unterstützt. Da er sich nicht auf eine Rolle festzulegen möchte, wechselt er zwischen der Leitung von Projekten, der Programmierung, der Gestaltung von Schnittstellen, dem Bau von Prototypen und dem Entwurf von Schaltkreisen. Bevor er zu IDEO kam, erwarb er an der Brown University einen BS in Elektrotechnik und belegte gleichzeitig Kurse an der Rhode Island School of Design, um seine kreativen Interessen zu vertiefen. Nebenbei arbeitet er an einer Reihe von analogen X-Y-Röhrenoszilloskopen im Eurorack-Format. Sie können ihn auf Instagram unter @ideo und @vondle_synths finden.

WEBLINKS

[1] Eyeo-Festival: https://eyeofestival.com/
[2] Demonstration mit Nitonol-Blumen: https://youtu.be/MBdbXO-WHJ4
[3] Adafruit Trinket M0: https://adafruit.com/product/3500
[4] Sparkfun Elektret-Mikrofon-Breakout: https://sparkfun.com/products/12758
[5] Muscle Wires Aktuatordraht 100 µm: https://elektor.link/musclewires
[6] Modifizierter Arduino-Bootloader: https://github.com/ideo/eyeo-flower/tree/main/Bootloader
[7] Eyeo-Flower Beispielsketch: https://github.com/ideo/ArduinoCore-samd/tree/master/libraries/Eyeo_Flower
[8] Arduino-Bibliothek Tone: https://arduino.cc/reference/en/libraries/tone/
[9] Mozzi: https://sensorium.github.io/Mozzi/
[10] Adafruit-Bibliothek Zero DMA: https://arduino.cc/reference/en/libraries/adafruit-zero-dma-library/
[11] Adafruit-Bibliothek Zero FFT: https://www.arduino.cc/reference/en/libraries/adafruit-zero-fft-library/
[12] Nitinol-Papierblüten-Prototyp (animiertes GIF): https://elektor.link/gif/nitinol-flower-paper-prototype
[13] YUPO, synthetisches Papier: https://yupousa.com/what-is-yupo/
[14] Adafruit-Bibliothek NeoPixel: https://arduino.cc/reference/en/libraries/adafruit-neopixel/
[15] EYEO-Flora, interaktive Seite: https://observablehq.com/@jftesser/eyeo-flower
[16] EYEO-Flower, GitHub-Repository: https://github.com/ideo/eyeo-flower
[17] Platinenhersteller JLCPCB: https://jlcpcb.com/
[18] Wie man SAMD-Bootloader programmiert: https://learn.adafruit.com/how-to-program-samd-bootloaders
[19] Atmel-ICE-Debugger, Benutzerhandbuch: https://elektor.link/AtmelICEUserGuide

Holen Sie sich die neue Arduino Hardware!

Es gibt nichts, was uns mehr begeistert, als neue Hardware in die Hände zu bekommen, und so war diese Zusammenarbeit mit Arduino ein wahrer Genuss! Möchten Sie sich davon überzeugen? Elektor hat das Lager erweitert, um alle Produkte, die in dieser Ausgabe vorgestellt werden, unterbringen zu können!

RP2040-gesteuerter Roboterarm Arduino Braccio++

Die nächste Evolution des Tinkerkit-Braccio-Roboters heißt Arduino Braccio ++, ein brandneuer Roboterarm, der für fortgeschrittene User entwickelt wurde. Arduino Braccio ++ kann auf verschiedene Arten zusammengebaut werden, um verschiedene Aufgaben zu erfüllen, beispielsweise das Bewegen von Objekten, das Verfolgen der eigenen Bewegungen mit einer angebrachten Kamera oder das Verfolgen der Bewegung der Sonne mit einem angebrachten Solarpanel. Arduino Braccio ++ bietet von Anfang an eine Vielzahl erweiterter Möglichkeiten, einschließlich eines neuen Braccio-Carriers mit LCD-Bildschirm, neuer RS-485-Servomotoren und einer völlig neuen Erfahrung.

www.elektor.de/20174

Praktische digitale PID-Steuerung mit Raspberry Pi und Arduino Uno

www.elektor.de/20274

Arduino Pro Nicla Vision

Nicla Vision vereint einen leistungsstarken STM32H747AII6 Dual ARM Cortex-M7/-M4 IC-Prozessor mit einer 2-MP-Farbkamera, die TinyML unterstützt, einem intelligenten 6-Achsen-Bewegungssensor, integriertem Mikrofon und einem Abstandssensor.

www.elektor.de/20152

Arduino Pro Portenta H7

Mit Portenta H7 können Sie Ihr nächstes intelligentes Projekt bauen. Wollten Sie schon immer ein automatisiertes Haus? Oder einen intelligenten Garten? Nun, mit den IoT-Cloud-kompatiblen Boards von Arduino ist das jetzt ganz einfach. Das bedeutet: Sie können Geräte verbinden, Daten visualisieren, Ihre Projekte von überall auf der Welt steuern und teilen.

www.elektor.de/19351

Arduino Pro Portenta X8

Portenta X8 ist ein leistungsstarkes, industrietaugliches SOM mit vorinstalliertem Linux-Betriebssystem, das dank seiner modularen Container-Architektur geräteunabhängige Software ausführen kann. Es handelt sich im Grunde um zwei Industrieprodukte in einem, die die Verfügbarkeit von Arduino-Bibliotheken und -Fähigkeiten mit einer Container-basierten Linux-Distribution kombinieren.

www.elektor.de/20270

Arduino Pro Nicla Sense ME

Ein neuer Standard für intelligente Sensoriklösungen.

www.elektor.de/20327

Portenta Vision Shield (Ethernet)

Das Portenta Vision-Shield bringt industrietaugliche Funktionen zu Ihrem Arduino-Portenta: professionelle Computer Vision, direktionale Audioerkennung, Ethernet und JTAG.

www.elektor.de/19511

Portenta Vision Shield LoRa®

Mit diesem Portenta-Hardware-Add-on können Sie eingebettete Computer-Vision-Anwendungen ausführen und sich drahtlos über LoRa® mit der Arduino-Cloud oder Ihrer eigenen Infrastruktur verbinden.

www.elektor.de/20332

Breakout Arduino Pro Portenta

Das Portenta Breakout-Board wurde entwickelt, um Hardware-Ingenieuren und -Herstellern bei der Entwicklung von Prototypen zu helfen und die Verbindungen und die Möglichkeiten der Geräte innerhalb der Portenta-Familie zu testen.

www.elektor.de/20341

Max-Carrier für Arduino Pro Portenta

Einfaches Prototyping Ihrer Portenta-Anwendungen, Einsatz in Nullzeit. Max Carrier verwandelt Portenta-Module in Einplatinencomputer oder Referenzdesigns, die KI für Hochleistungsanwendungen in der Industrie, Gebäudeautomatisierung und Robotik ermöglichen.

www.elektor.de/20271

Arduino Uno Rev3

Der klassische AVR-Mikrocontroller mit hoher Leistung und geringem Stromverbrauch. Der Uno ist das beste Board für den Einstieg in die Elektronik und das Programmieren. Der Uno ist einfach das robusteste Board, um mit der Arduino-Plattform zu basteln.

www.elektor.de/15877

Arduino Make-Your-Uno Kit

Ein neuer Bausatz mit einem DIY-though-Hole-UNO und allen Bauteilen, um deinen UNO aufzubauen und einen eigenen UNO-gesteuerten Synthesizer zu bauen!

www.elektor.de/20330

Arduino Ethernet-Shield 2

www.elektor.de/19941

Arduino Sensor-Kit Basis

www.elektor.de/19944

Arduino Nano

Der Arduino Nano ist ein kleines, komplettes und Breadboard-steckbares Board, das auf dem ATmega328 basiert und im kleinsten verfügbaren Formfaktor von 18x45 mm verpackt ist!

www.elektor.de/17002

Arduino Nano RP2040 Connect

Das Arduino Nano RP2040 Connect ist ein RP2040-basiertes Arduino-Board mit WLAN, Bluetooth, einem Mikrofon und einem sechsachsigen intelligenten Bewegungssensor mit KI-Funktionen.

www.elektor.de/19754

Arduino Nano 33 BLE Sense

Nutzen Sie die Macht der KI mit dem leistungsstärkeren nRF52840-Prozessor und einer Reihe von eingebetteten Sensoren und der Möglichkeit, Edge-Computing-Anwendungen (KI) auszuführen.

www.elektor.de/19936

Unterstützung durch **Arduino Reseller**

Diese Arduino-Gastausgabe von Elektor wurde durch die Unterstützung folgender Mitglieder der Arduino-Reseller-Community ermöglicht.

Besuchen Sie sie, wenn Sie Arduino-Produkte benötigen!

GOTRON AALST GENT HASSELT

www.gotron.be

HELLAS digital

www.hellasdigital.gr

TINYTRONICS

www.tinytronics.nl

Paradisetronic.com

www.paradisetronic.com

Techni Science.

www.techniscience.com

reichelt elektronik – The best part of your project

www.reichelt.com/arduino

WHADDA MADE BY VELLEMAN

www.whadda.com

KUBII

www.kubii.fr

GOTRONIC ROBOTIQUE ET COMPOSANTS ÉLECTRONIQUES

www.gotronic.fr